

### **CPX-E-EP Integration with Omron Sysmac PLC through EtherNet/IP communication**

A brief explanation of the this document describes about Integration of Festo CPX-E-EP Expansion Modules ( IO-Link Master's, Digital Input's, Digital Output's, Analog Inputs' & Analog Output's ) with Omron Sysmac PLC via Ethernet-IP Protocol.

It also describes about the Integration of Festo CPX-E-EP Modules with Omron NJ301 Series PLC.

CPX-E-EP

Title .....CPX-E-EP Integration with Omron Sysmac PLC through EtherNet/IP Communication  
Version ..... 1.10  
Document no. .... 100335  
Original .....en  
Author .....Festo  
Last saved ..... 13.04.2021

## Copyright Notice

This documentation is the intellectual property of Festo SE & Co. KG, which also has the exclusive copyright. Any modification of the content, duplication or reprinting of this documentation as well as distribution to third parties can only be made with the express consent of Festo SE & Co. KG.

Festo SE & Co KG reserves the right to make modifications to this document in whole or in part. All brand and product names are trademarks or registered trademarks of their respective owners.

## Legal Notice

Hardware, software, operating systems and drivers may only be used for the applications described and only in conjunction with components recommended by Festo SE & Co. KG.

Festo SE & Co. KG does not accept any liability for damages arising from the use of any incorrect or incomplete information contained in this documentation or any information missing therefrom.

Defects resulting from the improper handling of devices and modules are excluded from the warranty.

The data and information specified in this document should not be used for the implementation of safety functions relating to the protection of personnel and machinery.

No liability is accepted for claims for damages arising from a failure or functional defect. In other respects, the regulations with regard to liability from the terms and conditions of delivery, payment and use of software of Festo SE & Co. KG, which can be found at [www.festo.com](http://www.festo.com) and can be supplied on request, shall apply.

All data contained in this document do not represent guaranteed specifications, particularly with regard to functionality, condition or quality, in the legal sense.

The information in this document serves only as basic information for the implementation of a specific, hypothetical application and is in no way intended as a substitute for the operating instructions of the respective manufacturers and the design and testing of the respective application by the user.

The operating instructions for Festo products can be found at [www.festo.com/sp](http://www.festo.com/sp).

Users of this document (application note) must verify that all functions described here also work correctly in the application. By reading this document and adhering to the specifications contained therein, users are also solely responsible for their own application.

# Table of Content

|          |                                                                              |           |
|----------|------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction .....</b>                                                    | <b>7</b>  |
| 1.1      | Hardware's Used .....                                                        | 7         |
| 1.2      | Software's Used .....                                                        | 7         |
| <b>2</b> | <b>Application Description .....</b>                                         | <b>8</b>  |
| 2.1      | Hardware Setup .....                                                         | 8         |
| 2.2      | Communication Architecture .....                                             | 9         |
| 2.3      | Hardware Over View of CPX-E-EP Bus Module .....                              | 10        |
| 2.4      | Display Components .....                                                     | 11        |
| 2.5      | Expansion Module Connection .....                                            | 12        |
| <b>3</b> | <b>Hardware Configuration .....</b>                                          | <b>13</b> |
| 3.1      | CPX-E-EP Module Network IP Address Configuration Using Rotary Switches ..... | 13        |
| 3.2      | CPX-E-EP Module Network IP Address Configuration Using FDT .....             | 14        |
| 3.3      | CPX-E-EP Module Network IP Address Configuration Using FAS .....             | 16        |
| 3.4      | CPX-E-EP Diagnostic DIL Switch Configuration .....                           | 19        |
| 3.5      | Analog Input Module Hardware Configuration .....                             | 20        |
| 3.6      | Analog Output Module Hardware Configuration .....                            | 23        |
| 3.7      | IO-Link Module Hardware Configuration .....                                  | 24        |
| 3.8      | Start Up Module Configuration Parameters.....                                | 26        |
| <b>4</b> | <b>Parameter Setup.....</b>                                                  | <b>27</b> |
| 4.1      | CIPUCMMSend .....                                                            | 27        |
| 4.1.1    | Variables of CIPUCMMSend .....                                               | 27        |
| 4.1.2    | Data Type.....                                                               | 28        |
| 4.1.3    | Function .....                                                               | 28        |
| 4.1.4    | Data type of RqPath .....                                                    | 28        |
| 4.1.5    | Sample Programming for Parameter Read .....                                  | 30        |
| 4.1.6    | Sample Programming for Parameter Write.....                                  | 31        |
| <b>5</b> | <b>Sysmac Studio Configuration .....</b>                                     | <b>33</b> |
| 5.1      | EDS File Installation.....                                                   | 33        |
| 5.2      | Starting a Project.....                                                      | 36        |
| 5.3      | PLC EtherNet/IP Address Configuration .....                                  | 36        |
| 5.4      | EtherNet/IP Target Device Variable Configuration .....                       | 38        |
| 5.5      | EtherNet/IP Target Device Configuration .....                                | 40        |
| 5.5.1    | Adding the Target Device .....                                               | 40        |
| 5.5.2    | Register the Tag Set for Target Device .....                                 | 42        |
| 5.5.3    | Target Device Input / Output Connection .....                                | 43        |

|          |                                                    |           |
|----------|----------------------------------------------------|-----------|
| 5.6      | PLC Configuration Transfer to Controller .....     | 43        |
| <b>6</b> | <b>Addressing Concept .....</b>                    | <b>47</b> |
| 6.1      | Addressing of IO-Link Modules.....                 | 48        |
| 6.2      | Sample Code for IO-Link Module .....               | 49        |
| 6.3      | Addressing of Digital Input Modules .....          | 50        |
| 6.4      | Sample Code for Digital Input Module.....          | 52        |
| 6.5      | Addressing of Digital Output Modules .....         | 54        |
| 6.6      | Sample Code for Digital Output Module .....        | 55        |
| 6.7      | Addressing of Analog Input Modules.....            | 56        |
| 6.8      | Sample Code for Analog Input Module .....          | 57        |
| 6.9      | Addressing of Analog Output Modules .....          | 58        |
| 6.10     | Sample Code for Analog Output Module .....         | 59        |
| <b>7</b> | <b>Diagnosis CPX-E-EP Module Errors .....</b>      | <b>61</b> |
| 7.1      | Addressing of Diagnostics Data.....                | 62        |
| 7.2      | I/O Diagnostics Interface.....                     | 62        |
| 7.3      | Organisation of Internal Data and Parameters ..... | 62        |
| 7.4      | Mode of Operation .....                            | 63        |
| 7.4.1    | Output Bits.....                                   | 63        |
| 7.4.2    | Input Bits .....                                   | 63        |
| 7.4.3    | Reading Out Diagnostic Data .....                  | 64        |
| 7.5      | Sample Code for Diagnostics Data Input .....       | 65        |
| 7.6      | Sample Code for Diagnostics Data Output .....      | 67        |
| <b>8</b> | <b>Help.....</b>                                   | <b>71</b> |





# 1 Introduction

This sample code uses digital Input, digital output, Analog input & Analog output modules from slot number 1 to 4 respectively in CPX-E-EP slave module. CPX-E-EP has the capacity to expand modules up to 10 numbers excluding bus module or controller. Also it has capability to transmit 64 Bytes inputs & 64 Bytes output data between controller and CPX-E module. This 64 Bytes Input Output size includes diagnostics data.

## 1.1 Hardware's Used

| Name                                         | Revision                  |
|----------------------------------------------|---------------------------|
| CPX-E-EP Remote Adapter                      | V 2.0 with EDS 02-05-2017 |
| Omron NJ301-1100 PLC                         | Ver. 1.18                 |
| CPX-E-4IOL IO-Link Master                    | V 2.0                     |
| CPX-E-16DI Digital Input Module              | V 1.0                     |
| CPX-E-8DO Digital Output Module              | V 1.0                     |
| CPX-E-4AI Analog Input Module                | V 2.0                     |
| CPX-E-4AO Analog Output Module               | V 1.0                     |
| EtherNet Cable                               | NEBC-R3G4-ES-1-S-R3G4-ET  |
| Power Cable for Interface & Expansion Module | Based on Application      |

Table 1.1: Hardware's Used

## 1.2 Software's Used

| Name                           | Firmware Version |
|--------------------------------|------------------|
| Festo Field Device Tool        | V 2.10.2.36790   |
| Festo Maintenance Tool         | V 4.21.210       |
| Festo Automation Suite         | V 2.1.1.1        |
| Sysmac Studio Standard Edition | Ver. 1.43        |

Table 1.1: Software's Used

## 2 Application Description

The main use of this application note is to establish the communication between Festo CPX-E-EP bus module and Omron PLC through EtherNet/IP protocol.

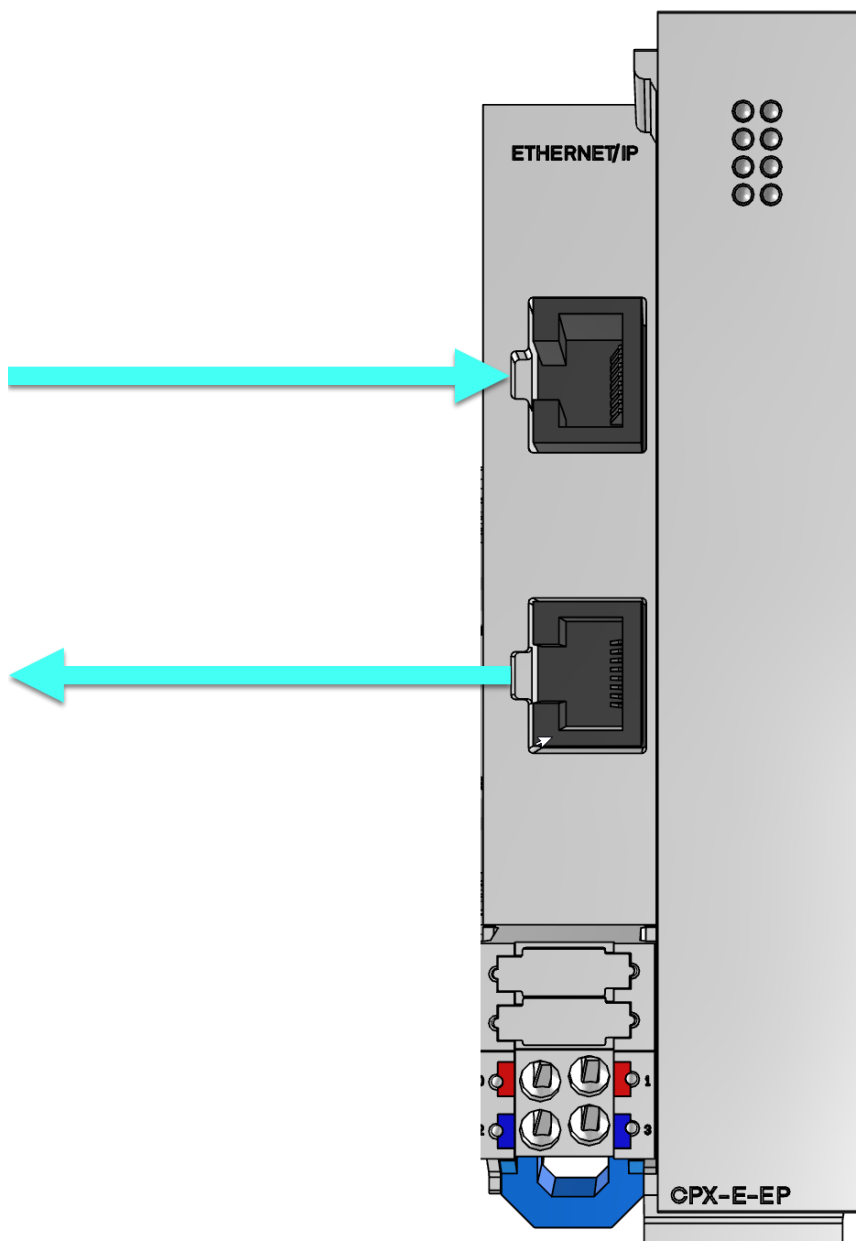
This application note explains in detail the contents of the **“EtherNet/IP Bus module Configuration, Parameter Settings & Diagnostics Data”** of CPX-E-EP Bus Module & its Expansion Modules integrating with Omron PLC via EtherNet/IP Protocol.

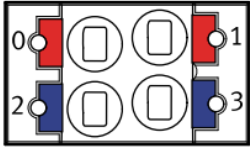
The application note has the description for the following:

- CPX-E-EP bus module and expansion units configuration.
- Target device ( CPX-E-EP ) configuration as EtherNet/IP device.
- Description of Communication between Omron Sysmac PLC & CPX-E-EP.

### 2.1 Hardware Setup

Connect the power supply of the head module labelled XD 0 & 2.



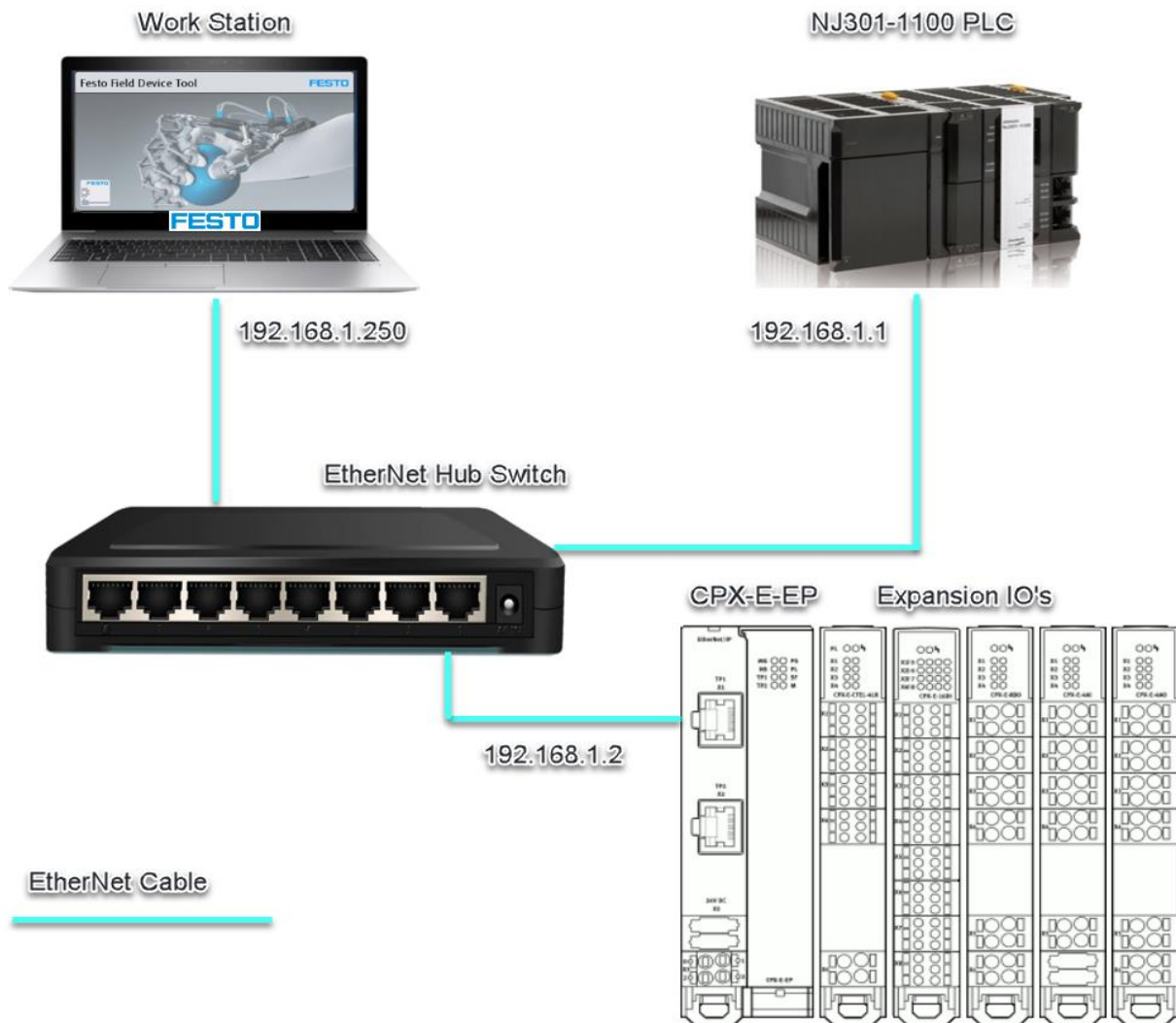
| Port [XD] <sup>1)</sup>                                                           | Signal                                       |
|-----------------------------------------------------------------------------------|----------------------------------------------|
|  | 0                                            |
|                                                                                   | 1                                            |
|                                                                                   | 2                                            |
|                                                                                   | 3                                            |
|                                                                                   | 0 V DC operating voltage supply $U_{EL/SEN}$ |

1) The ports XD.0 and XD.1 and also XD.2 and XD.3, respectively, are interconnected in the terminal strip.

Tab. 8 Port [XD]

## 2.2 Communication Architecture

When we want to communicate between Omron PLC and CPX-E-EP Bus Module via EtherNet/IP protocol the following system architecture must be followed.



## 2.3 Hardware Over View of CPX-E-EP Bus Module

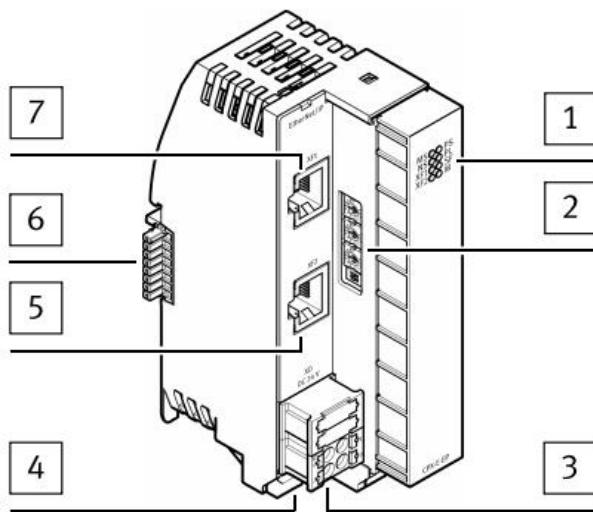


Fig. 2 Product design

- 1 LED displays
- 2 Rotary and DIL switches
- 3 Terminal strip for operating voltage supply  $U_{EL/SEN}$  [XD]
- 4 Terminal strip interlock
- 5 Network connection [XF2]
- 6 Linking element
- 7 Network connection [XF1]

## 2.4 Display Components

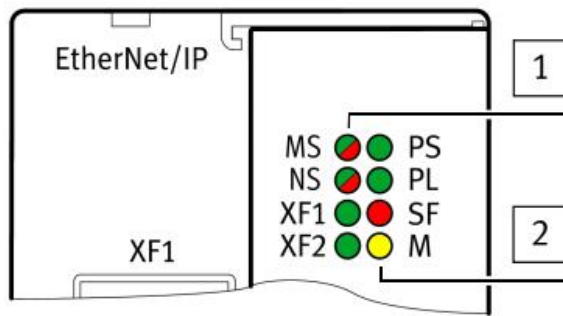
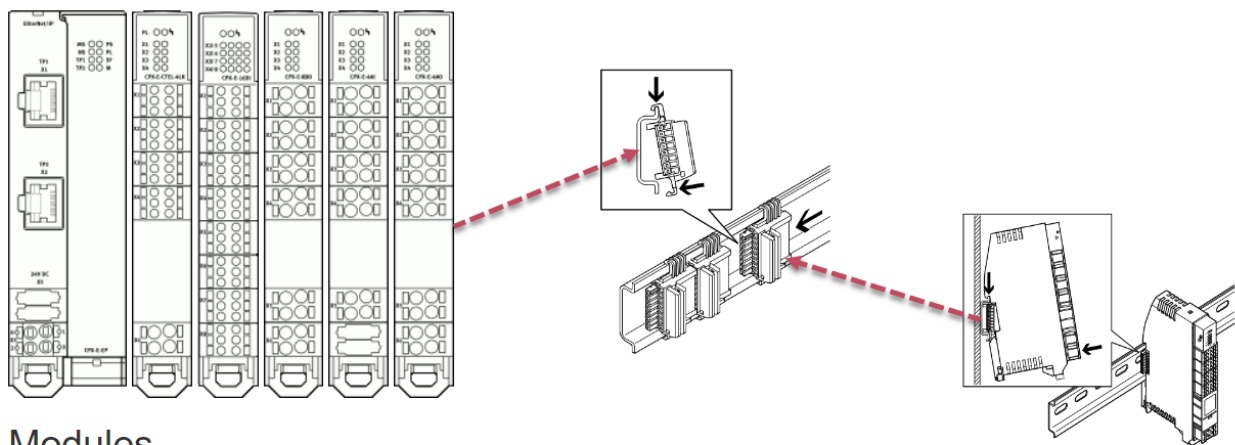


Fig. 3 LED displays

- 1 Network-specific LED displays:
  - Module status [MS] (green, red, orange)
  - Network status [NS] (green, red, orange)
  - Connection/data traffic [XF1]/[XF2] (green)
- 2 System-specific LED displays:
  - Operating voltage supply  $U_{EL/SEN}$  [PS] (green)
  - Load voltage supply  $U_{OUT}$  [PL] (green)
  - System error [SF] (rot)
  - Force mode [M] (yellow)

2.5 Expansion Module Connection



Modules

| Slot | Module    | Revision | Serial   | Diagnosis |
|------|-----------|----------|----------|-----------|
| 0    | E-EP      | 1        | 1F5A26A4 | OK        |
| 1    | E-4IOL    | 2        | DD485A3C | OK        |
| 2    | E-16DI    | 1        | DD47D80A | OK        |
| 3    | E-8DO     | 1        | DD47E231 | OK        |
| 4    | E-4AI-U-I | 2        | DD4761B4 | OK        |
| 5    | E-4AO-U-I | 1        | DD47B515 | OK        |

Make sure the Bus linking plugs are installed behind all modules. In case if the Bus linking plugs are not connected in expansion module after power ON, Module will indicate SF fault. IOLink, Analog Input & Output modules need external 24VDC supply.

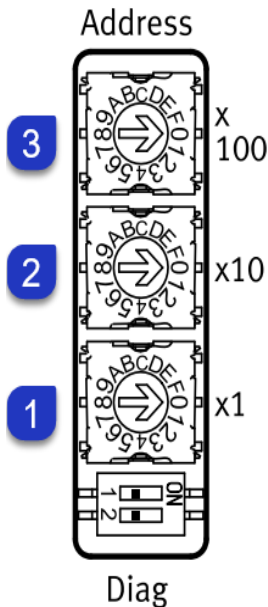
For more details about installation of expansion module refer from page number 6 to 9 in below manual.

[https://www.festo.com/cat/en-gb\\_gb/data/doc\\_ENGB/PDF/EN/CPX-E\\_EN.PDF](https://www.festo.com/cat/en-gb_gb/data/doc_ENGB/PDF/EN/CPX-E_EN.PDF)

### 3 Hardware Configuration

This chapter tells about Festo CPX-E-EP module network IP address, diagnostic bits enabling configuration, counter module's signal type configuration.

#### 3.1 CPX-E-EP Module Network IP Address Configuration Using Rotary Switches



Rotary switches from 1 to 3 is used to configure 4th octet value of the IP Address ( 192.168.1.xx). Rotary switches 1, 2 & 3 are represent 1's, 10's & 100's values of 4th octet value of the IP address.

Possible Settings:

- 0                      =>     Dynamic addressing via DHCP/BOOTP.
- 1 ... 255           =>     Permissible address range.

Valid Values:

- EtherNet/IP        =>     300 ... 555 (IP address = value – 300)
- Modbus TCP        =>     600 ... 855 (IP address = value – 600)
- Factory setting   =>     900

In case rotary switch setting is invalid, module automatically goes to dynamic ( DHCP ) mode.



#### Note

- A power re-start is mandatory to activate new configuration of bus module.
- For switch positions 0 ... 255, the protocols EtherNet/IP and Modbus TCP can be used in parallel. The rights to control the outputs are assigned to the protocol that sends outputs first.

For an example to set IP address of CPX-E-EP module as 192.168.1.2 rotary switch 1 must be in position 2, rotary switches 2 & 3 must be in position 0. Once all the configuration are done, open FESTO Field Device Tool software and ensure IP address is assigned as per the configuration. FESTO Field Device Tool download link is given in chapter – 4. Also make sure Omron PLC network IP address also same range as the CPX-E-EP module.

### 3.2 CPX-E-EP Module Network IP Address Configuration Using FDT

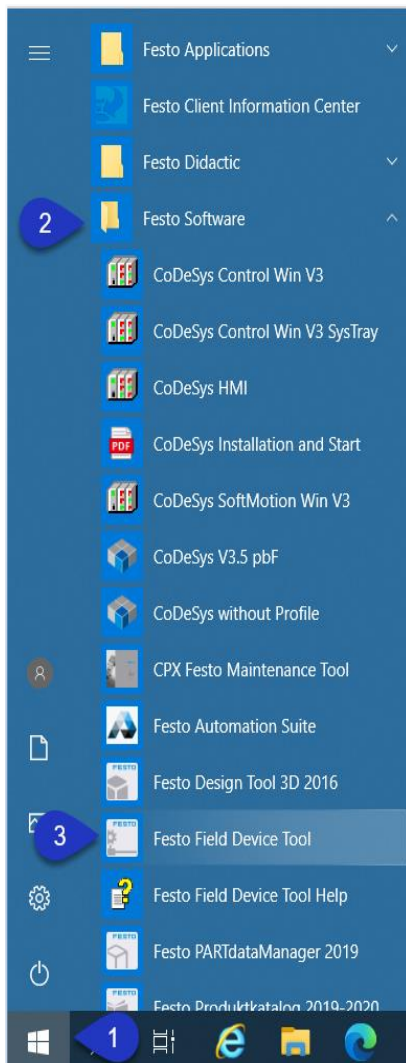
This chapter describes about IP address configuration of CPX-E-EP bus node parametrizing port using Festo Field Device Tool.



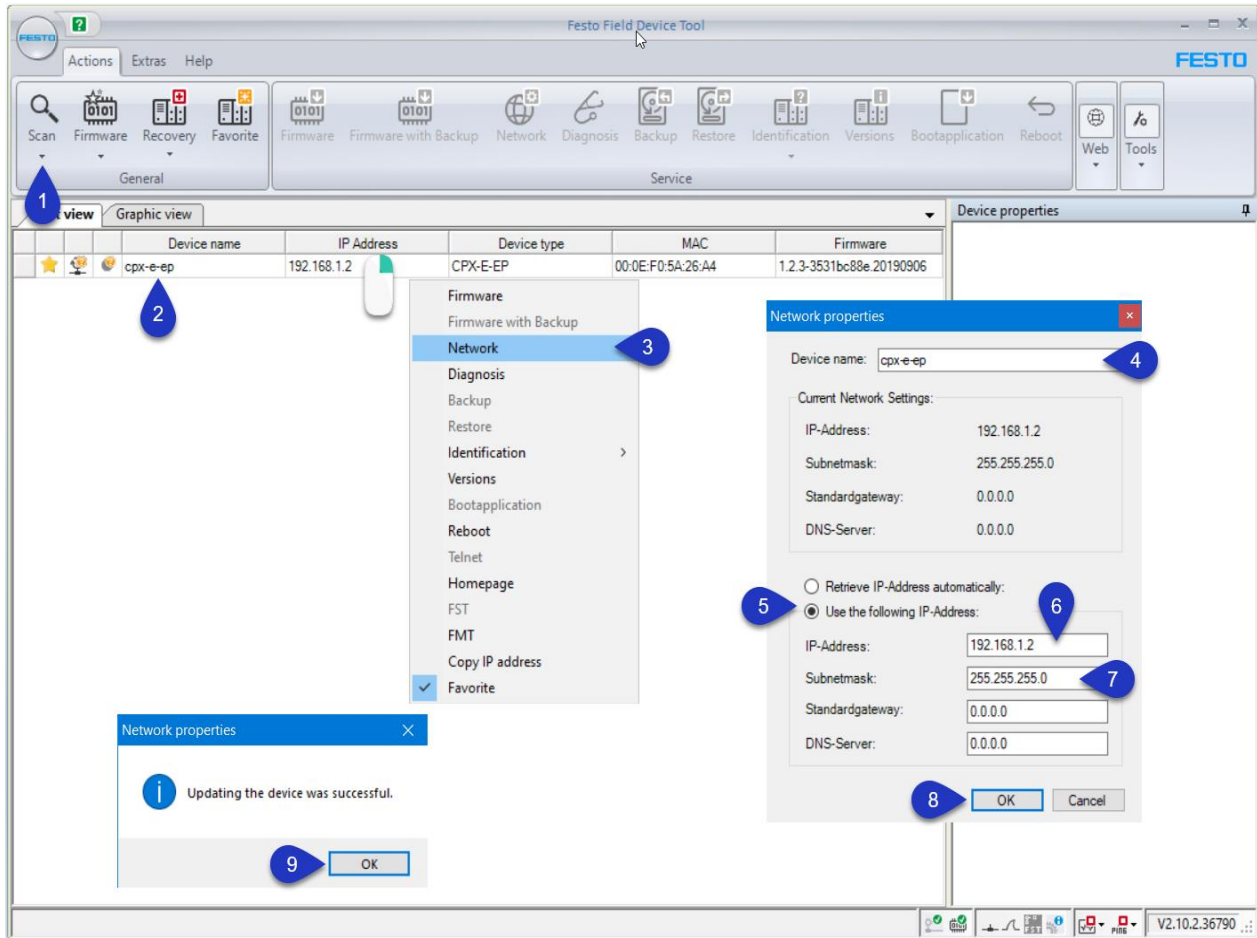
#### Note

- Before starting CPX-E-EP bus node IP address configuration process make sure the hardware connection between PC & bus node is done as per communication architecture shown in chapter 2.2.
- Make sure Rotary Switches setting is 0 (Dynamic addressing via DHCP/BOOTP).

#### Step 1



1. Click **“Windows Start”** button.
2. Go to **“Festo Software”** folder.
3. Select **“Festo Field Device Tool”** software.

**Step 2**

1. Select “**Scan Button**” to update the connected device list.
2. Select the correct bus node from device list based on identifiers like Device name & MAC address of devices.
3. Right click the selected device & click the “**Network**” option from right click menu.
4. Enter the “**New name**” for the drive. It’s an optional point by default drive name will be device type name.
5. Select “**Use following IP-Address**” option for enable static IP mode. By default the IP address mode will in Dynamic mode.
6. Enter the **new IP address** for the bus node. Make sure the drive IP series must meet series of PLC IP address (for this sample code use 192.168.1.2).
7. Enter the **subnet mask** value to define capacity of the network range. Above network configuration done for 255 devices.
8. Click “**OK**” to confirm configuration.
9. Click “**OK**” button from pop-up window to confirm the reboot the bus node.

**Step 3**

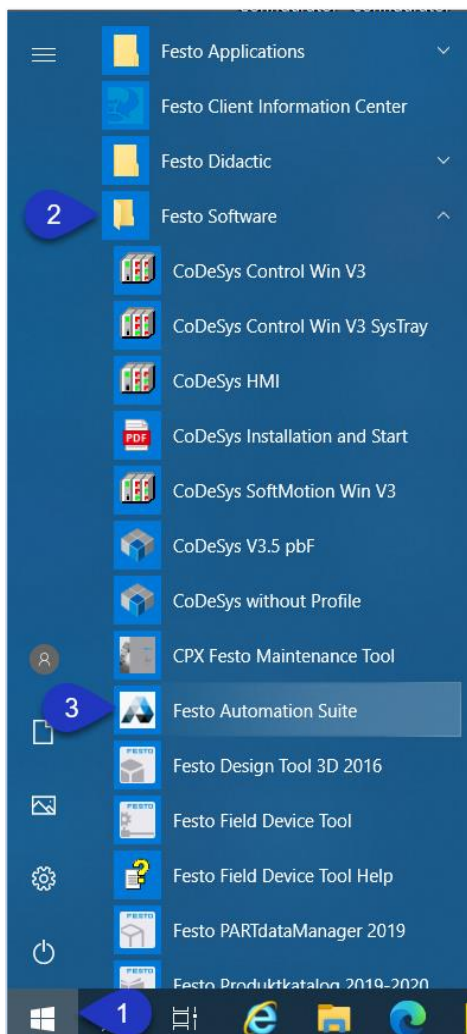
| After IP Address Configuration |   |              |             |             |                   |                     |
|--------------------------------|---|--------------|-------------|-------------|-------------------|---------------------|
| List view                      |   | Graphic view |             |             |                   |                     |
|                                |   | Device name  | IP Address  | Device type | MAC               | Firmware            |
|                                | ★ | cpx-e-ep     | 192.168.1.2 | CPX-E-EP    | 00:0E:F0:5A:26:A4 | 1.2.3-3531bc88e.201 |

### 3.3 CPX-E-EP Module Network IP Address Configuration Using FAS

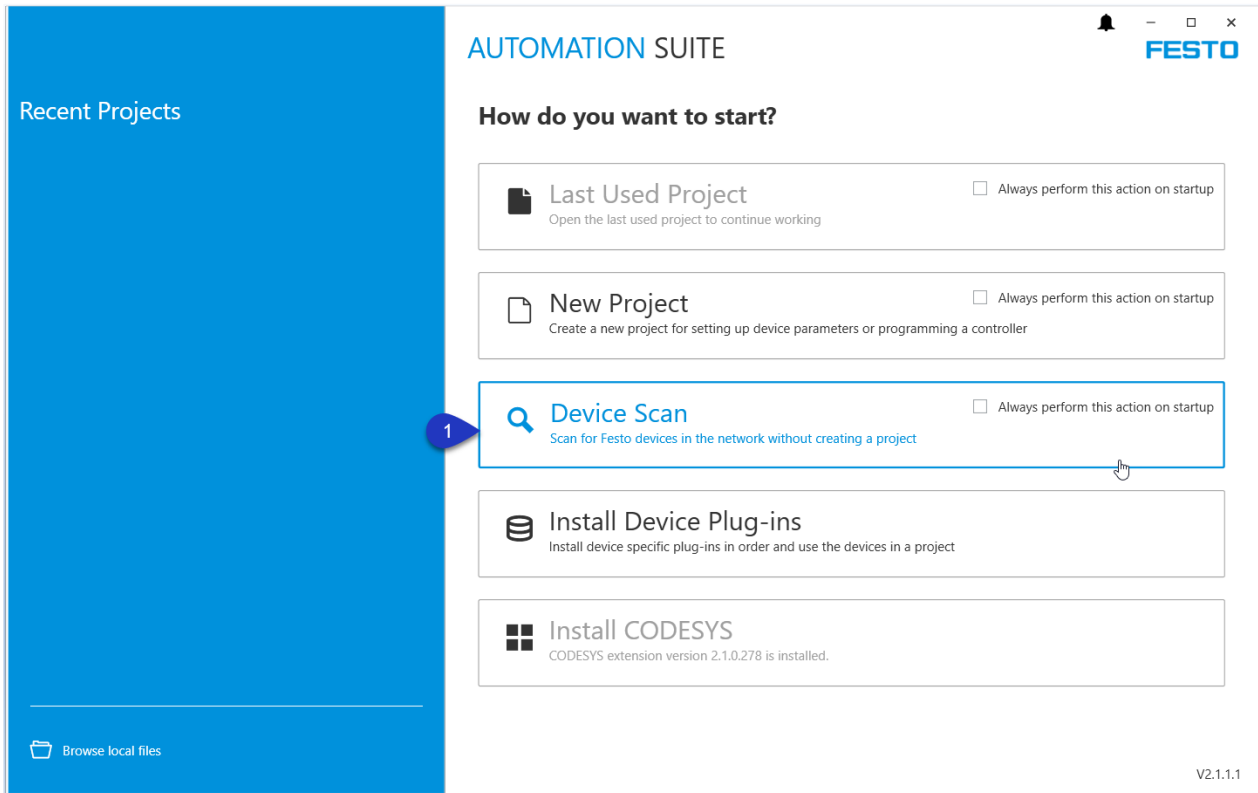
This chapter describes about IP address configuration of CPX-E-EP parametrizing port using Festo Automation Suite.

**Note**

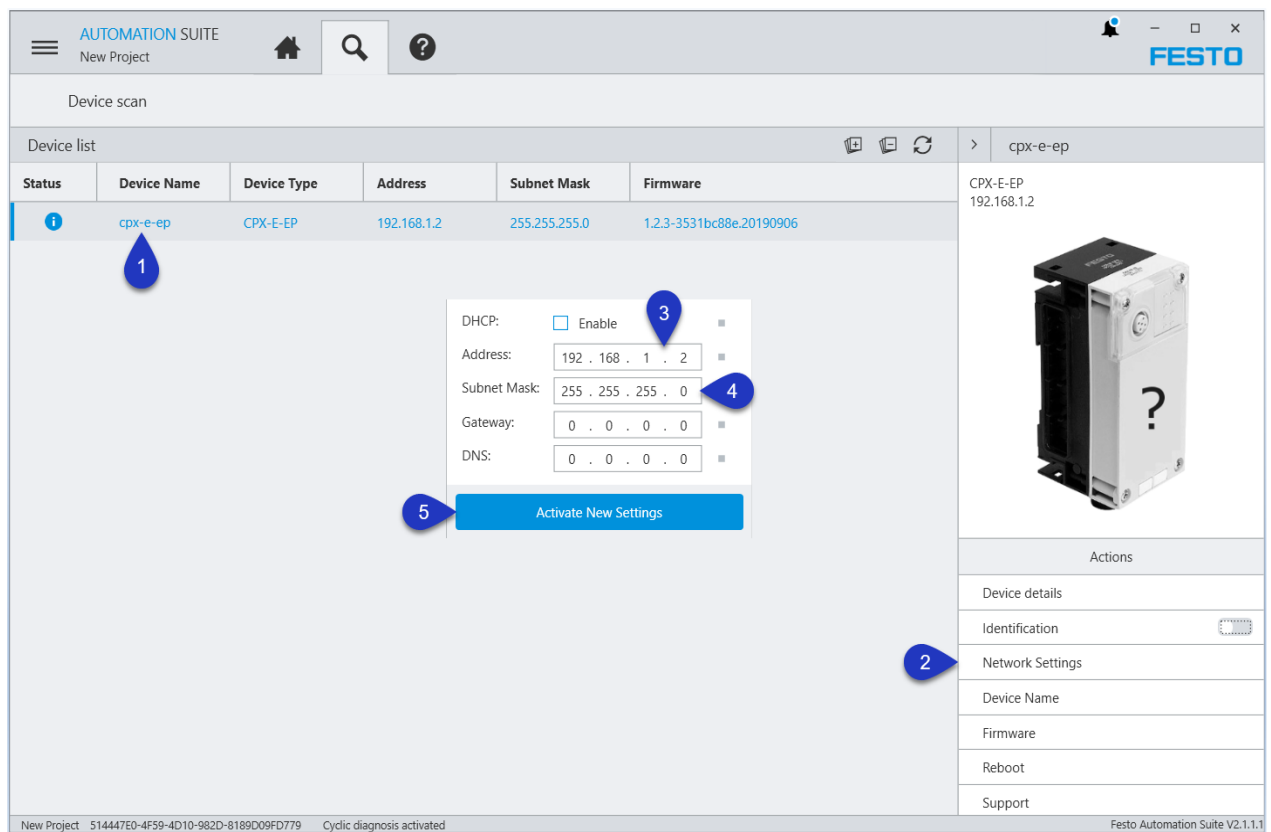
- Before starting CPX-E-EP bus node IP address configuration process make sure the hardware connection between PC & bus node is done as per communication architecture shown in chapter 2.2.
- Make sure Rotary Switches setting is 0 (Dynamic addressing via DHCP/BOOTP).

**Step 1**

1. Click **“Windows Start”** button.
2. Go to **“Festo Software”** folder.
3. Select **“Festo Automation Suite”** software.

**Step 2**

1. Click **“Device Scan”** from home screen to scan for Festo devices in the network.

**Step 3**

1. Select correct device by its name.
2. Select “**Network Setting**” from action window.
3. Enter the “**New IP address**” for the bus node. Make sure the drive IP series must meet series of PLC IP address ( for this sample code use 192.168.1.2 ).
4. Enter the **subnet mask** value to define capacity of the network range. Above network configuration done for 255 devices.
5. Finally click “**Activate New Setting**” to update the changes.

#### Step 4

### Network Settings

The device might reboot automatically when the new settings are activated!



### Network settings updated successfully

Updating the device was successful. The device reboots automatically if necessary and changes will take effect immediately.



1. Click “**OK**” button from pop-up window to confirm the reboot the bus node.
2. Click “**OK**” button from pop-up window to confirm the successful update.



#### Note

- Please follow any one chapter 3.1 to 3.3 for configure CPX-E-EP bus node IP address configuration

### 3.4 CPX-E-EP Diagnostic DIL Switch Configuration

| DIL switch                                                                        | Function                 |                                    |
|-----------------------------------------------------------------------------------|--------------------------|------------------------------------|
|  | DIL 1: OFF<br>DIL 2: OFF | No diagnostics <sup>1)</sup>       |
|  | DIL 1: OFF<br>DIL 2: ON  | Status bits activated              |
|  | DIL 1: ON<br>DIL 2: OFF  | I/O diagnostic interface activated |
|  | DIL 1: ON<br>DIL 2: ON   | Reserved                           |

1) Factory setting

This sample code is programmed with I/O diagnostic interface active mode. So DIL switch – 1 must be in ON position & switch – 2 must be in OFF position as highlighted in above table. When I/O diagnostic is enabled it occupies first 16 bit inputs & 16 bit output from IO's array . If user needs different configuration based on above table it can be possible.

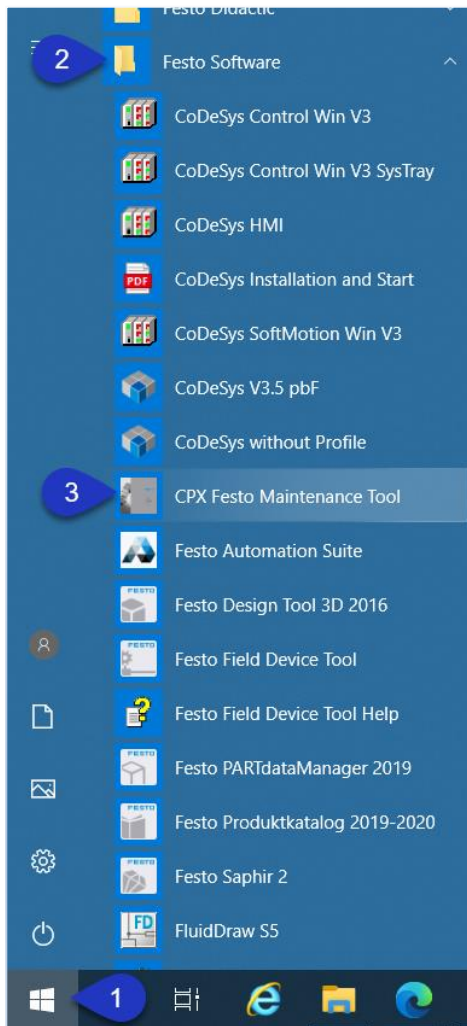


#### Note

- Changes to the rotary and DIL switches only take effect following a restart of the bus module.
- In case DIL switch is selected for status bits active mode it occupies first 8 bits from Input tags & no tag occupied in output tags. In this mode only status of module can able to read ( Function :1936)
- In case DIL switch is selected for I/O diagnostic enable mode the various error can be able to handle & parameter read/write will be possible form controller.

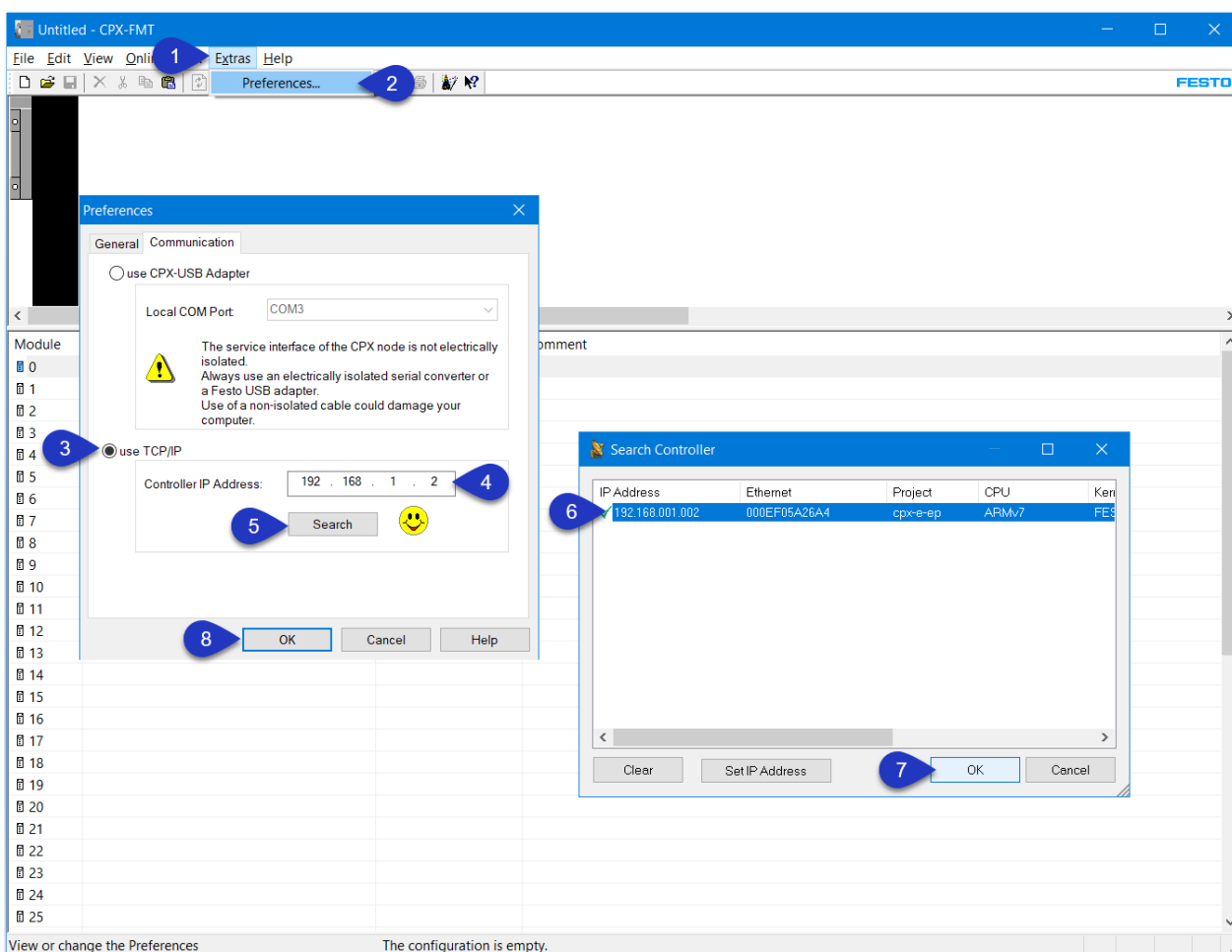
### 3.5 Analog Input Module Hardware Configuration

#### Step 1

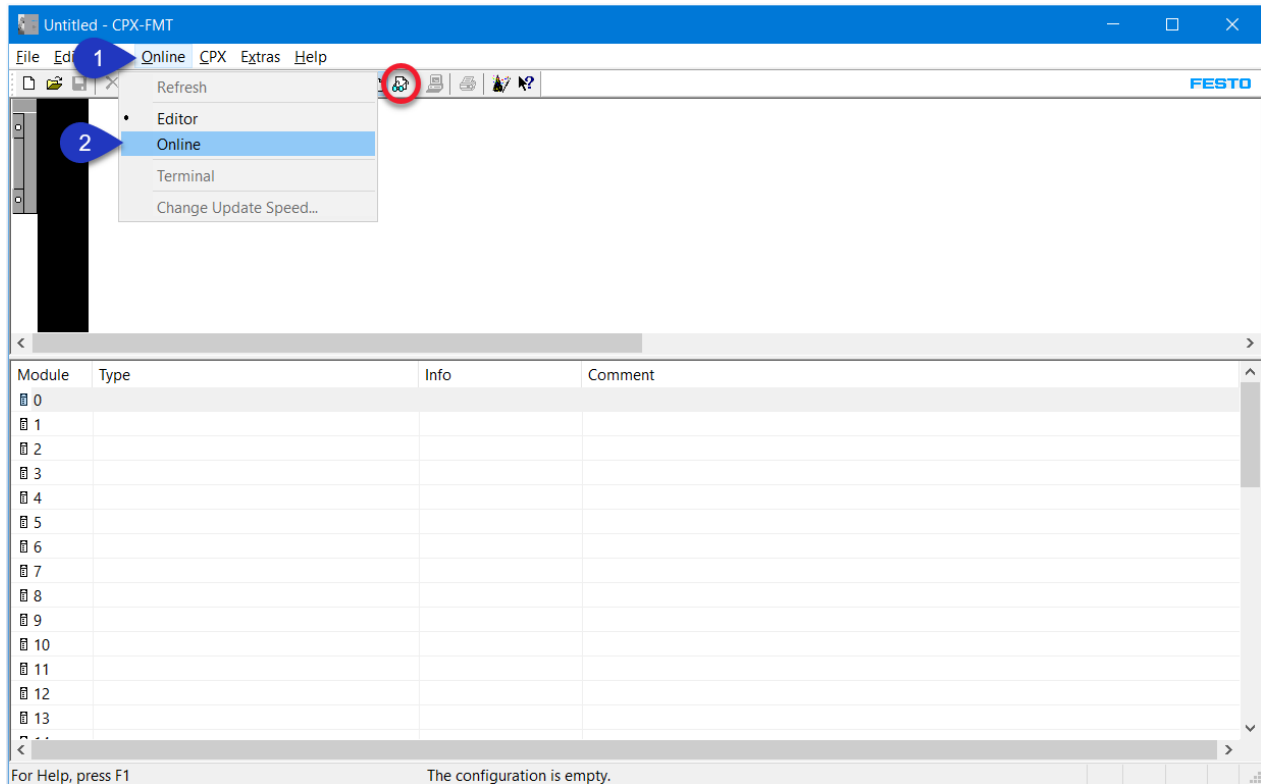


1. Click **“Windows Start”** button.
2. Go to **“Festo Software”** folder.
3. Select **“CPX Festo Maintenance Tool ”** software.

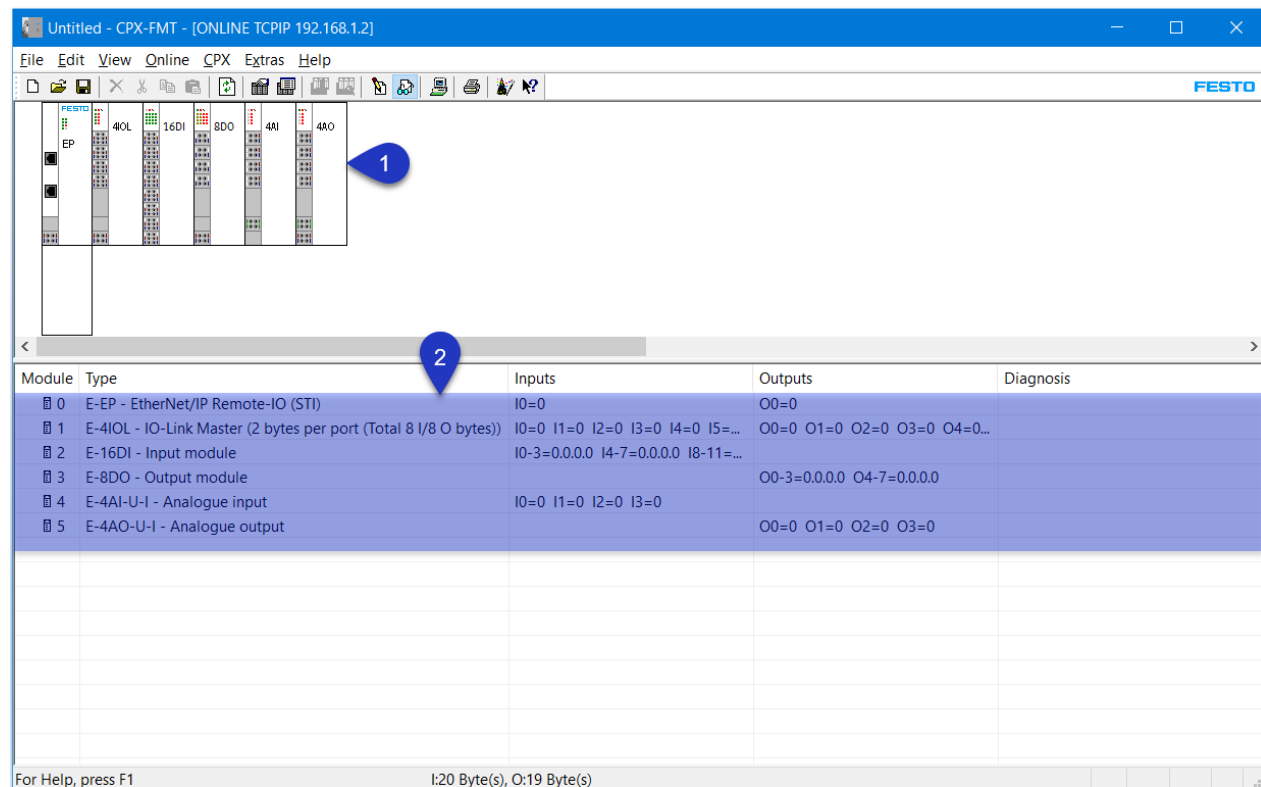
#### Step 2



1. Open “CPX-FMT” software and select “Extras” from the toolbar
2. Select “Preferences” from extras menu.
3. Select “Use TCP/IP” communication option from preferences menu.
4. Enter the “Controller IP Address” of the connected controller or bus node ( for this sample code use 192.168.1.2 ).
5. Click “Search” button for launch search controller window.
6. Once search controller window opens, It automatically list IP address connected in network. “Select hardware IP address from list to establish communications”.
7. Click “OK” from search controller window to ensure the selected address.
8. Click “OK” button from preferences.

**Step 3**

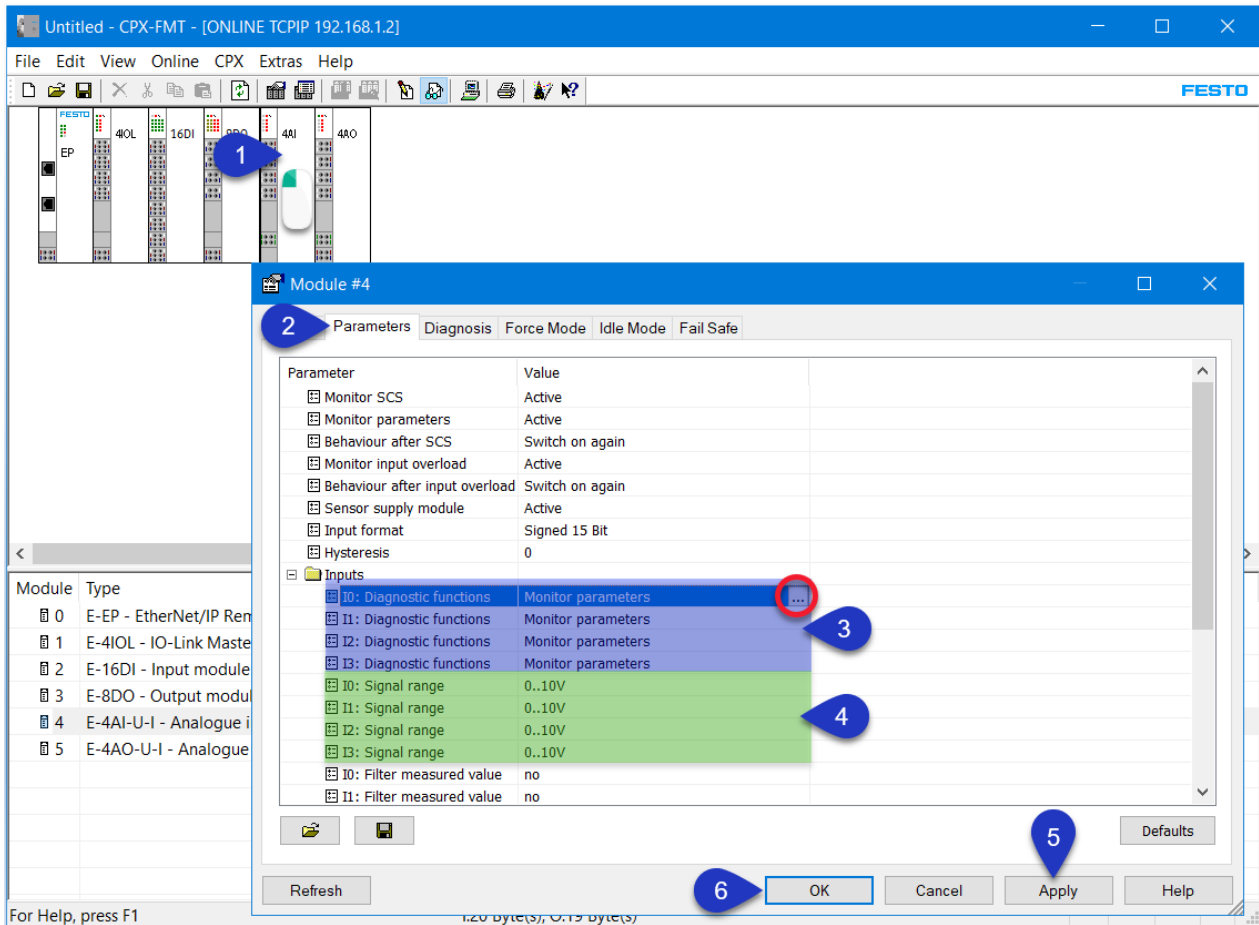
1. Select **"Online"** from tool bar. User can also use the shortcut button shown in red circle to go online.
2. Select **"Online"** from online menu.

**Step 4**

1. Connected controller or bus node with its expansion modules on network in graphic view.

- Input, output and diagnosis view of each module on network.

### Step 5



- Select Analog input module and double click to open module configuration window.
- Select **"Parameter"** tab from module configuration window.
- Expand Inputs parameter folder by clicking plus symbol in folder. Blue highlighted parameters are monitor parameter enable/disable for each channel of inputs.
- Green highlighted parameters are signal range selection parameters for each channels.
- After all the channels configuration click "Apply" button.
- Click "OK" to finalize the configurations.

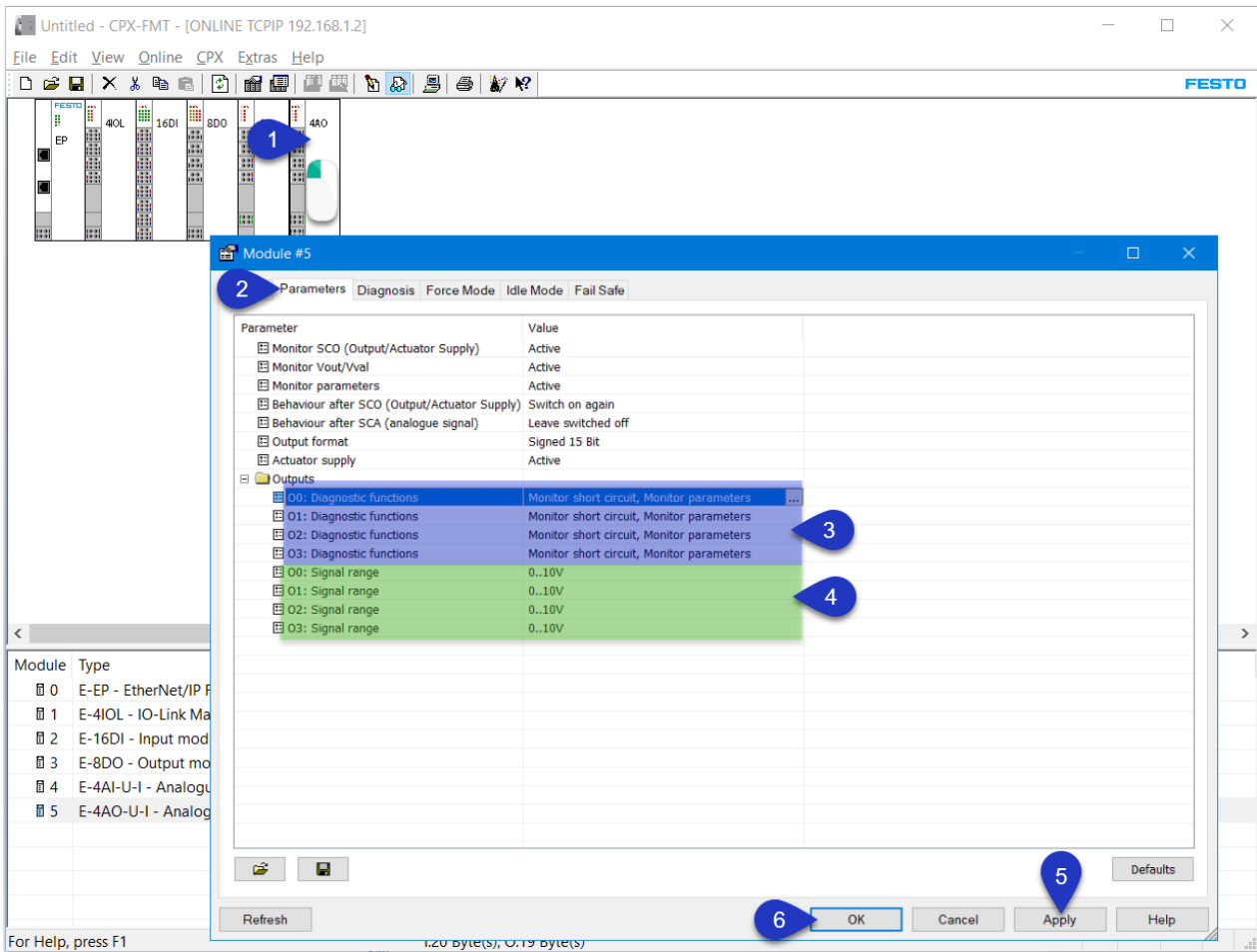
## 3.6 Analog Output Module Hardware Configuration

This topic explains about CPX-E Analog output module basic configurations.

### Step 1

If user is not online with CPX-E-EP module follow the chapter 3.5 step – 2 & 3 to go online. If online connection is established jump to step – 2.

## Step 2



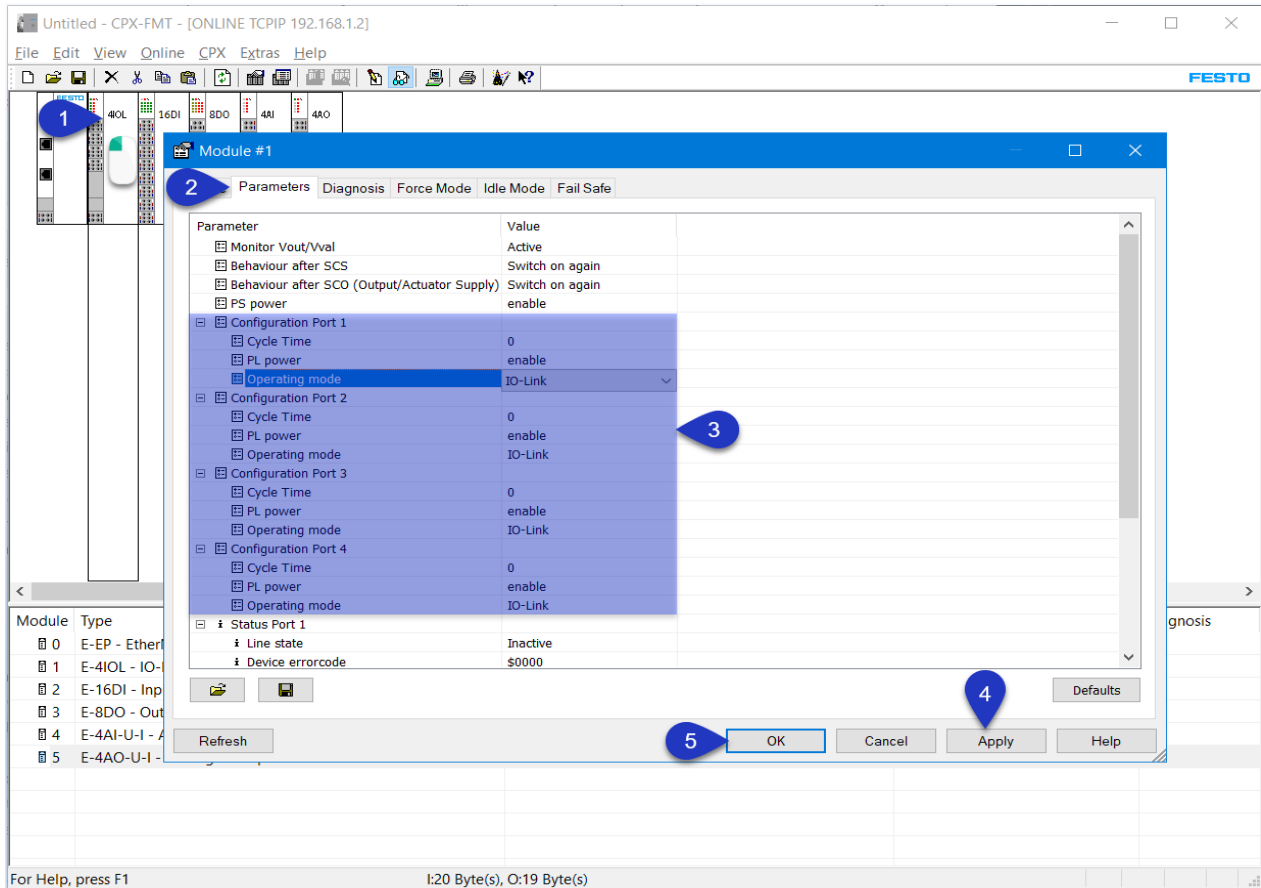
1. Select a Analog output module and double click to open configuration window.
2. Select **"Parameters"** tab from configuration window.
3. Expand Outputs parameter folder by clicking plus symbol in folder. Blue highlighted parameters are monitor parameter enable/disable for each channel of outputs.
4. Green highlighted parameters are signal range selection parameters for each channels.
5. After all the channels configuration click **"Apply"** button.
6. Click **"OK"** to finalize the configurations.

### 3.7 IO-Link Module Hardware Configuration

This topic explains about CPX-E IO-Link module basic configurations.

#### Step 1

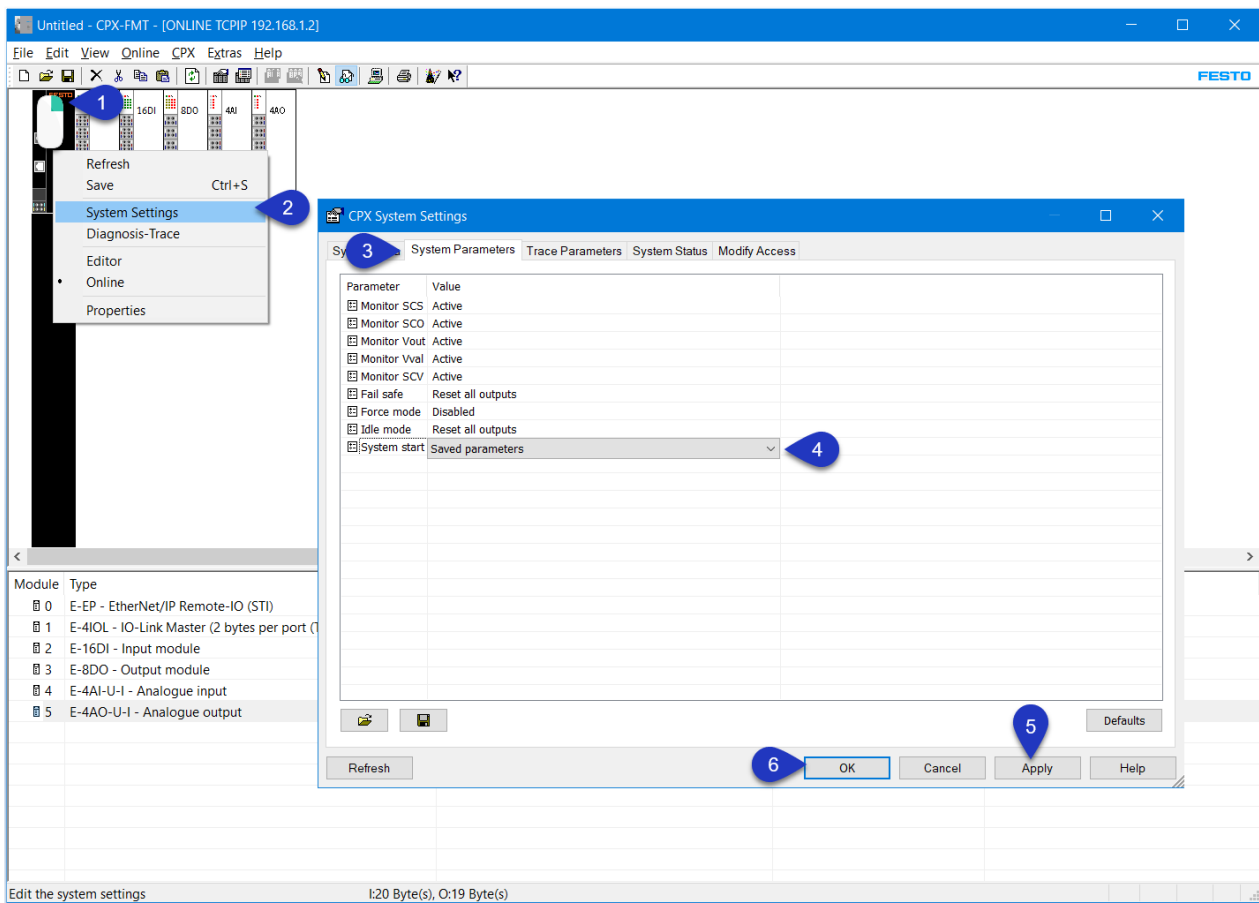
If user is not online with CPX-E-EP module follow the chapter 3.5 step – 2 & 3 to go online. If online connection is established jump to step – 2.

**Step 2**

1. Select a IO-Link module and double click to open configuration window.
2. Select **"Parameters"** tab from configuration window.
3. Expand Configuration Port 1 to 4 folder by clicking plus symbol in folder. Blue highlighted parameters are operating mode for each channel of ports.
4. After all the channels configuration click "Apply" button.
5. Click "OK" to finalize the configurations.

For Digital input & output modules factory default parameters is enough to get a input channel feedback & to set output channels. if user needs to do more configuration like input debounce time & signal extend time follow same procedure as Analog and IO-Link modules to reach parameters tab.

### 3.8 Start Up Module Configuration Parameters



1. Select a Bus node module and right click to open menu window.
2. Select **"System Settings"** option from menu.
3. Select **"Saved Parameters"** from drop down menu of system start.
4. After all the channels configuration click **"Apply"** button.
5. Click **"OK"** to finalize the configurations.

Once all the modules configurations are done, save the configuration and download the configuration of CPX-EEP by clicking download  button in FMT software.



#### Note

- After configuration change power re-start is mandatory to activate new configuration in bus module.

After power restart user can ensure configuration details as required by double clicking CPX-E-EP module in FMT software. General configuration details of CPX-E-EP will be displayed in module tab of CPX-E-EP property window.

## 4 Parameter Setup

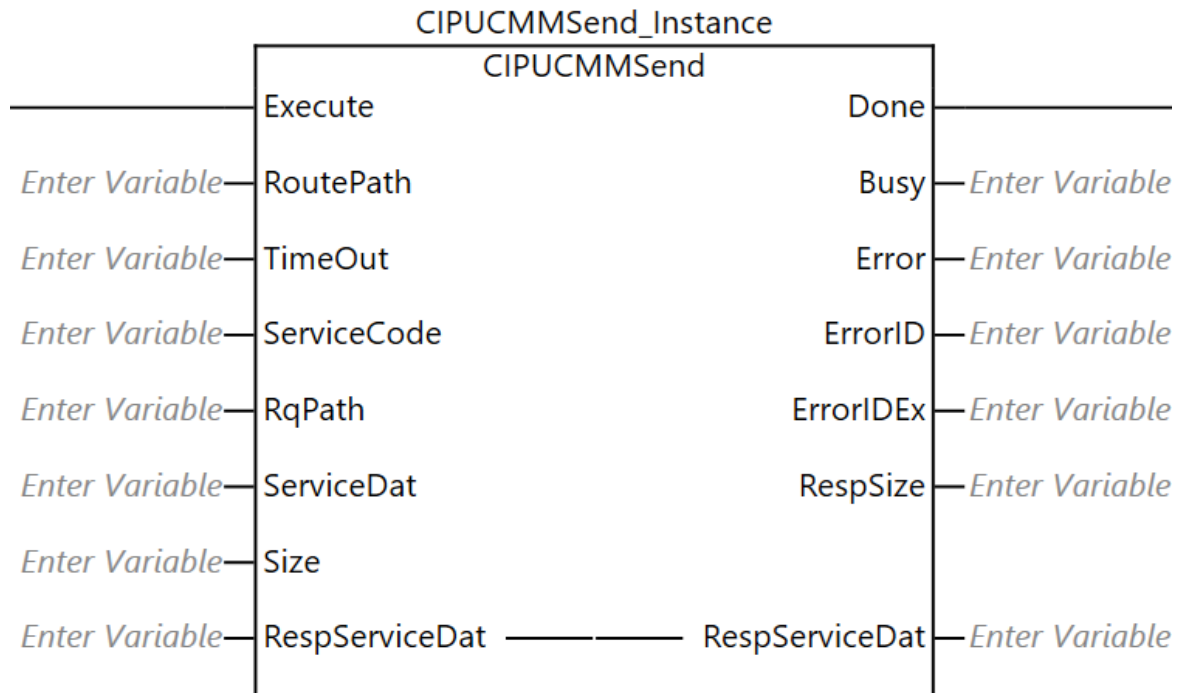
In Omron Sysmac PLCs EtherNet slave device parameters are read / write through following function blocks namely,

- CIPUCMMSSend

### 4.1 CIPUCMMSSend

The CIPUCMMSSend instruction sends a UCMM CIP message to a specified device on a CIP network.

The below image shows the schematic view of the Function Block.



#### 4.1.1 Variables of CIPUCMMSSend

| Parameter      | Meaning                    | I/O                        | Description        | Valid Range          | Data Type | Default    |
|----------------|----------------------------|----------------------------|--------------------|----------------------|-----------|------------|
| RoutePath      | Route path                 | Input                      | Route path         | Depends on data type | ---       | ---        |
| TimeOut        | Timeout time               |                            | Timeout time       | 1 to 65535           | 0.1 s     | 20 (2.0 s) |
| ServiceCode    | Service code               |                            | Service code       | Depends on data type | ---       | ---        |
| RqPath         | Request path               |                            | Request path       | ---                  |           | ---        |
| ServiceDat     | Command data               |                            | Data to send       | Depends on data type |           | *1         |
| Size           | Number of elements to send | Number of elements to send | 1                  |                      |           |            |
| RespServiceDat | Response data              | In-Out                     | Response data      |                      | ---       |            |
| RespSize       | Response size              | Output                     | Response data size |                      | Bytes     | ---        |

Table 4.1.1: Variables of CIPUCMMSSend



#### Note

\*1 If you omit an input parameter, the default value is not applied. A building error will occur.

### 4.1.2 Data Type

|                | Boolean                                                                                                                                  | Bit strings |      |       |       | Integers |      |       |       |      |     |      |      | Real numbers |       | Times, durations, dates, and text strings |      |     |    |        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|-------------|------|-------|-------|----------|------|-------|-------|------|-----|------|------|--------------|-------|-------------------------------------------|------|-----|----|--------|
|                | BOOL                                                                                                                                     | BYTE        | WORD | DWORD | LWORD | USINT    | UINT | UDINT | ULINT | SINT | INT | DINT | LINT | REAL         | LREAL | TIME                                      | DATE | TOD | DT | STRING |
| RoutePath      |                                                                                                                                          |             |      |       |       |          |      |       |       |      |     |      |      |              |       |                                           |      |     |    | OK     |
| TimeOut        |                                                                                                                                          |             |      |       |       |          | OK   |       |       |      |     |      |      |              |       |                                           |      |     |    |        |
| ServiceCode    |                                                                                                                                          | OK          |      |       |       |          |      |       |       |      |     |      |      |              |       |                                           |      |     |    |        |
| RqPath         | Structure <code>_sREQUEST_PATH</code> or <code>_sREQUEST_PATH_EX</code> <sup>1)</sup> .<br>Refer to <a href="#">CIPSend</a> for details. |             |      |       |       |          |      |       |       |      |     |      |      |              |       |                                           |      |     |    |        |
| ServiceDat     |                                                                                                                                          | OK          | OK   | OK    | OK    | OK       | OK   | OK    | OK    | OK   | OK  | OK   | OK   | OK           | OK    |                                           |      |     |    |        |
| Size           |                                                                                                                                          |             |      |       |       |          | OK   |       |       |      |     |      |      |              |       |                                           |      |     |    |        |
| RespServiceDat |                                                                                                                                          | OK          | OK   | OK    | OK    | OK       | OK   | OK    | OK    | OK   | OK  | OK   | OK   | OK           | OK    |                                           |      |     |    |        |
| RespSize       |                                                                                                                                          |             |      |       |       |          | OK   |       |       |      |     |      |      |              |       |                                           |      |     |    |        |

<sup>1)</sup> A CPU Unit with unit version 1.11 or later and Sysmac Studio version 1.15 or higher are required to specify `_sREQUEST_PATH_EX` type.

### 4.1.3 Function

- The CIPUCMMSend instruction sends command data ServiceDat for the service specified with service code ServiceCode as a UCMM explicit message.
- The destination is specified with route path RoutePath.
- RqPath specifies the request path.
- Size specifies the number of elements to send.
- If ServiceDat is an array, specify the number of elements to send.
- If ServiceDat is not an array, always specify 1.
- If no service data is required, set Size to 0.
- The response data received later is stored in RespServiceDat. The number of bytes of the response data is stored in RespSize.
- TimeOut specifies the timeout time. If a response does not return within the timeout time, it is assumed that communications failed.
- The data type of ClassIDLogicalFormat, InstanceIDLogicalFormat, and AttributeIDLogicalFormat is enumerated type `_eCIP_LOGICAL_FORMAT`.

The meanings of the enumerators of enumerated type `_eCIP_LOGICAL_FORMAT` are as follows:

| Enumerator          | Meaning |
|---------------------|---------|
| <code>_8BIT</code>  | 8 bits  |
| <code>_16BIT</code> | 16 bits |
| <code>_32BIT</code> | 32 bits |



#### Note

- If the value of ErrorID is WORD#16#1C00, the CIP message error code is stored in ErrorIDEx.
- The meaning and values of ErrorIDEx depend on the remote node. Refer to the manual for the remote node.

### 4.1.4 Data type of RqPath

- The data type of RqPath is structure `_sREQUEST_PATH` or `_sREQUEST_PATH_EX`.
- Normally, use `_sREQUEST_PATH`.
- When you specify any logical format size, use `_sREQUEST_PATH_EX`.

**\_sREQUEST\_PATH Type**

| Name          | Meaning         | Description                                               | Data Type      | Valid Range          | Unit | Default |
|---------------|-----------------|-----------------------------------------------------------|----------------|----------------------|------|---------|
| RqPath        | Request path    | Request path                                              | _sREQUEST_PATH | ---                  | ---  | ---     |
| ClassID       | ClassID         | ClassID                                                   | UINT           | Depends on data type | ---  | 0       |
| InstanceID    | InstanceID      | InstanceID                                                | UINT           |                      |      | False   |
| isAttributeID | Attribute usage | True : Attribute ID used<br>False : Attribute ID not used | BOOL           |                      |      | 0       |
| AttributeID   | AttributeID     | AttributeID                                               | UINT           |                      |      |         |

Table 4.1.4: Table 1 \_sREQUEST\_PATH Type Specification

**\_sREQUEST\_PATH\_EX Type**

| Name                     | Meaning                     | Description                                               | Data Type            | Valid Range          | Unit | Default |
|--------------------------|-----------------------------|-----------------------------------------------------------|----------------------|----------------------|------|---------|
| RqPath                   | Request path                | Request path                                              | _sREQUEST_PATH_EX    | ---                  | ---  | ---     |
| ClassIDLogicalFormat     | Class ID logical format     | Class ID data size                                        | _eCIP_LOGICAL_FORMAT | Depends on data type | ---  | _8BIT   |
| ClassID                  | Class ID                    | Class ID                                                  | UDINT                |                      |      | 0       |
| InstanceIDLogicalFormat  | Instance ID logical format  | Instance ID data size                                     | _eCIP_LOGICAL_FORMAT |                      |      | _8BIT   |
| InstanceID               | Instance ID                 | Instance ID                                               | UDINT                |                      |      | 0       |
| isAttributeID            | Attribute usage             | True : Attribute ID used<br>False : Attribute ID not used | BOOL                 |                      |      | False   |
| AttributeIDLogicalFormat | Attribute ID logical format | Attribute ID data size                                    | _eCIP_LOGICAL_FORMAT |                      |      | _8BIT   |
| AttributeID              | Attribute ID                | Attribute ID                                              | UDINT                |                      |      | 0       |

Table 4.1.4: Table 2 \_sREQUEST\_PATH\_EX Type Specification

**Note**

- The logical format size of each ID in \_sREQUEST\_PATH type is 16 bits.
- If ServiceDat or RespServiceDat is an array, pass a subscripted array element to it as the parameter.
- User can read a maximum of 492 bytes of data. Refer Instruction Manual for more info.

#### 4.1.5 Sample Programming for Parameter Read


**Note**

To read / write parameter or send message through CIP UCMM from controller make start up module configuration parameters as default at CPX-FMT. Refer Chapter 3.8 for brief.



- This sample uses CIP UCMM messages to write a variable, read a variable, and send a message.
- The Controllers are connected to an EtherNet/IP network. The IP address of the controller is 192.168.1.1 and remote node is 192.168.1.2.
- The CIPUCMMSend instruction is used to read an explicit message from a remote node.

The following sample code used for to read parameter from remote node module slot 1 – IO-Link module port 1 and uses following EtherNet/IP objects.

| Remote Node Objects | PLC Function Block Object | Value |
|---------------------|---------------------------|-------|
| Object Class        | ClassID                   | 140   |
| Instances           | InstanceID                | 2     |
| ---                 | isAttributeID             | True  |
| Attribute No        | AttributeID               | 1     |


**Note**

Counting mode

- For the module-orientated objects, the following applies:
  - Instance number = Module number + 1
- The modules are counted starting at 0 for the bus module.
- The instances are counted starting at 1 for the bus module.

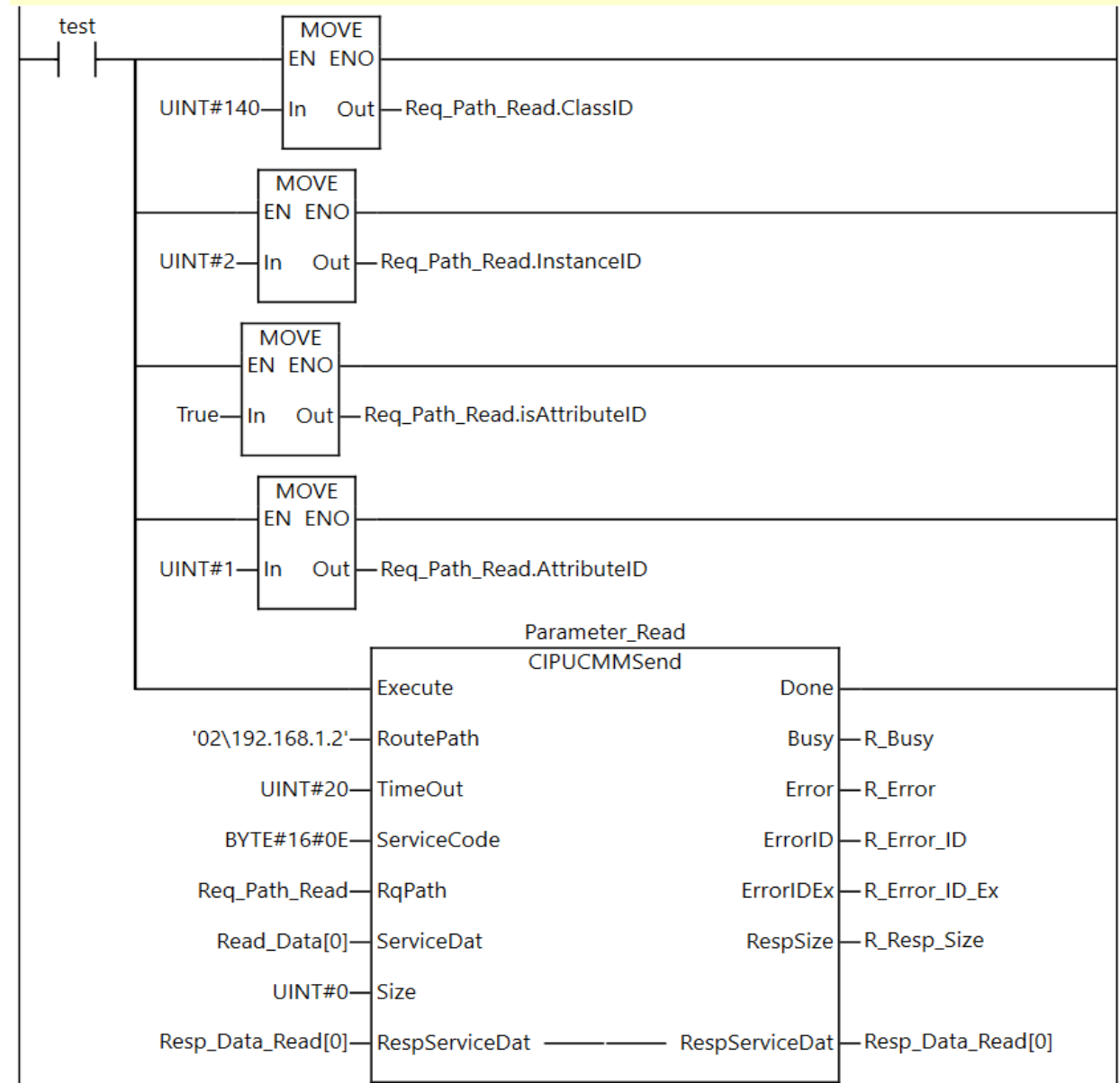
#### EtherNet/IP Class Services

The automation system CPX-E supports various services depending on the object:

| Service code          | Service name         |
|-----------------------|----------------------|
| 05 (05 <sub>h</sub> ) | Reset                |
| 01 (01 <sub>h</sub> ) | Get Attribute All    |
| 14 (0E <sub>h</sub> ) | Get Attribute Single |
| 16 (10 <sub>h</sub> ) | Set Attribute Single |

Tab. 12 EtherNet/IP Class Services

## CPX-E-EP Remote Terminal Units Parameter Read Function



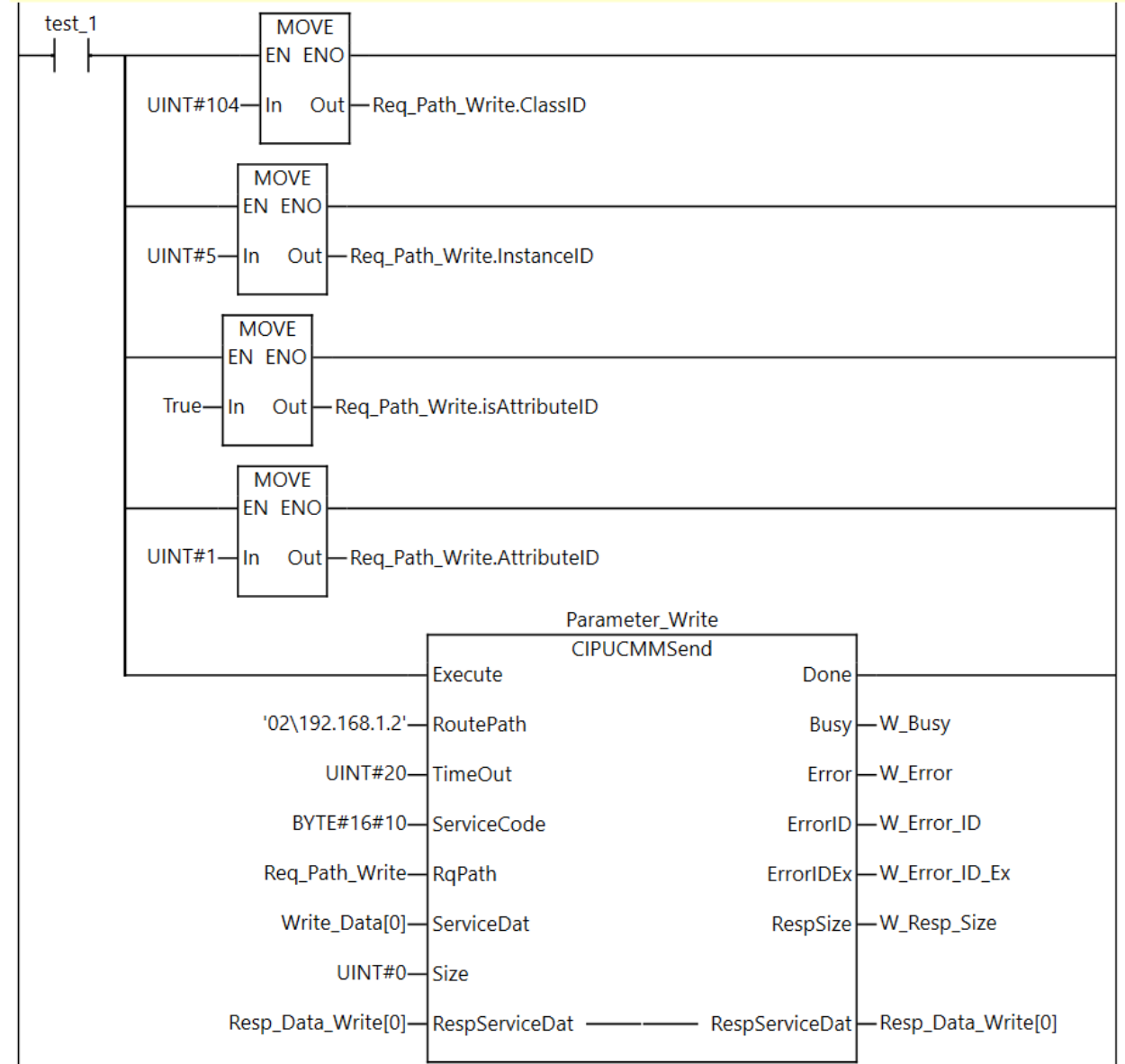
#### 4.1.6 Sample Programming for Parameter Write

- This sample uses CIP UCMM messages to write a variable, read a variable, and send a message.
- The Controllers are connected to an EtherNet/IP network. The IP address of the controller is 192.168.1.1 and remote node is 192.168.1.2.
- The CIPUCMMSend instruction is used to send an explicit message to a remote node.

The following sample code used for to write parameter to remote node module slot 4 – Analog Input module Channel 1 and uses following EtherNet/IP objects.

| Remote Node Objects | PLC Function Block Object | Value |
|---------------------|---------------------------|-------|
| Object Class        | ClassID                   | 104   |
| Instances           | InstanceID                | 5     |
| ---                 | isAttributeID             | True  |
| Attribute No        | AttributeID               | 1     |

## CPX-E-EP Remote Terminal Units Parameter Write Function

**Note**

- For more information about CPX-E-EP remote node EtherNet/IP parameterization please refer manual **“CPX-E-EP-EN Bus node - EtherNet/IP - Function - Modbus TCP – Parameterisation”** chapter 3.
- Link to the manual has been provided at help section.

## 5 Sysmac Studio Configuration

This chapter describes software configurations like CPX-E-EP EDS file import, EtherNet/IP for Omron Sysmac PLC Configuring, Adding Target Device.

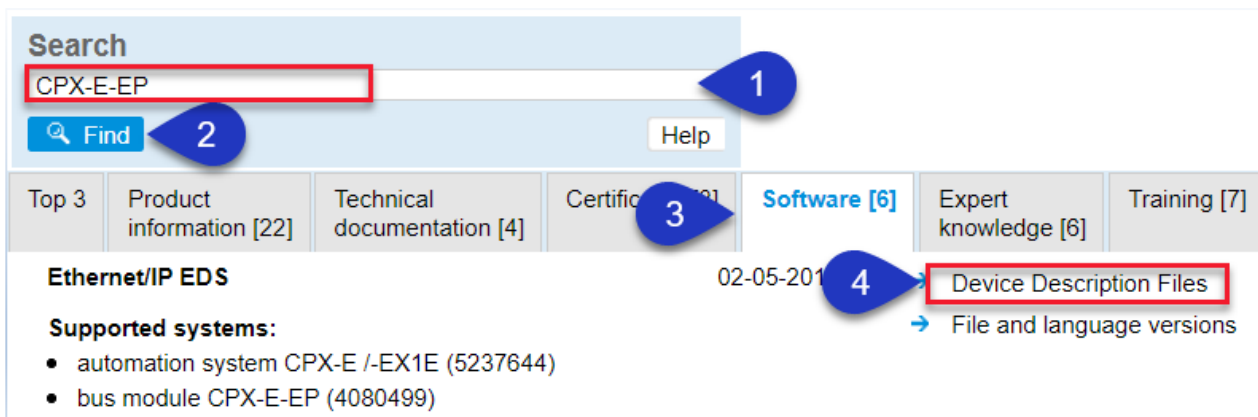
This chapter gives explanation about PLC program configuration for CPX-E-EP integration, Analog module scaling for engineering unit, Digital module address mapping & error diagnostic example for CPX-E-EP module.

### 5.1 EDS File Installation

#### Step 1

Download the ESI file from the Festo Support Portal on Support / Downloads page.

[https://www.festo.com/net/en-in\\_in/SupportPortal/default.aspx?q=CPX-E-EP&tab=4](https://www.festo.com/net/en-in_in/SupportPortal/default.aspx?q=CPX-E-EP&tab=4)



1. Type “CPX-E-EP” at search bar.
2. Click “Find” button.
3. Go to “Software” tab.
4. Click “Device Description File” to download EDS file.

#### Step 2



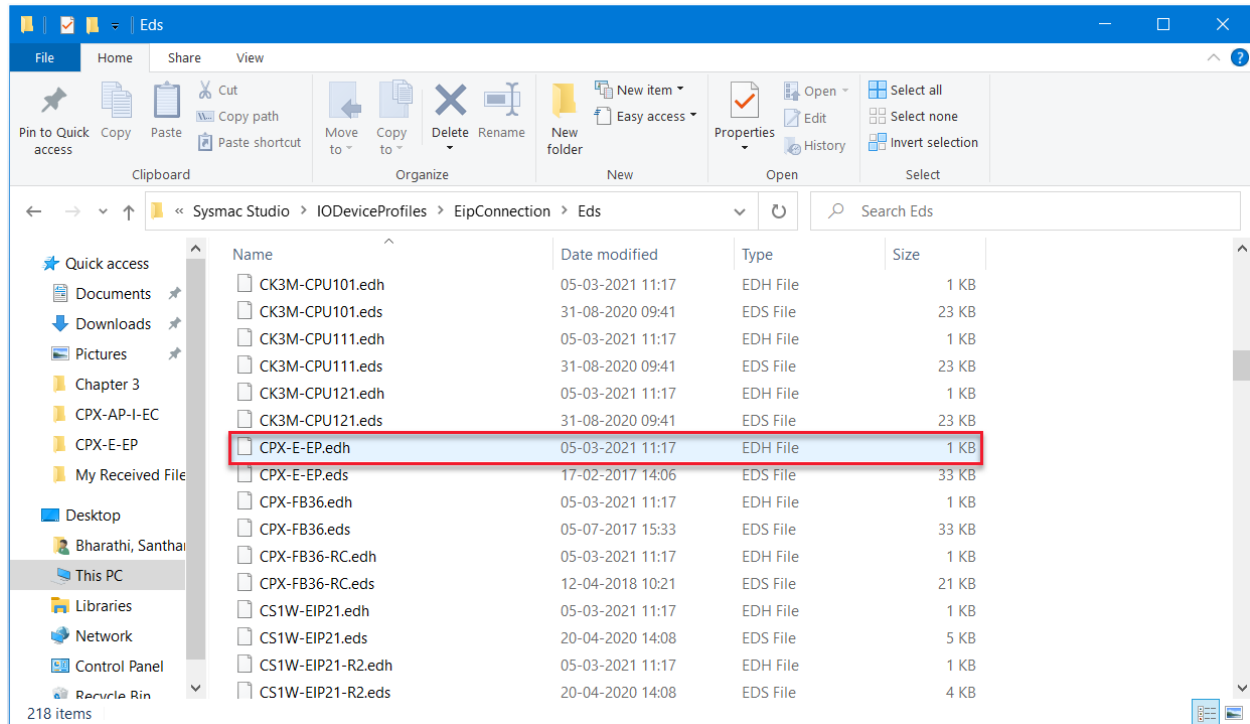
#### Note

EDS file for the Target Device can be installed by two methods on Sysmac Studio Software. Please use either one method as per user PC condition.

#### Method 1

1. Make sure that Omron Sysmac Studio is closed.
2. Download the EtherNet/IP EDS File for CPX-E-EP from the Festo Support Portal.
3. Paste it into the following folder.

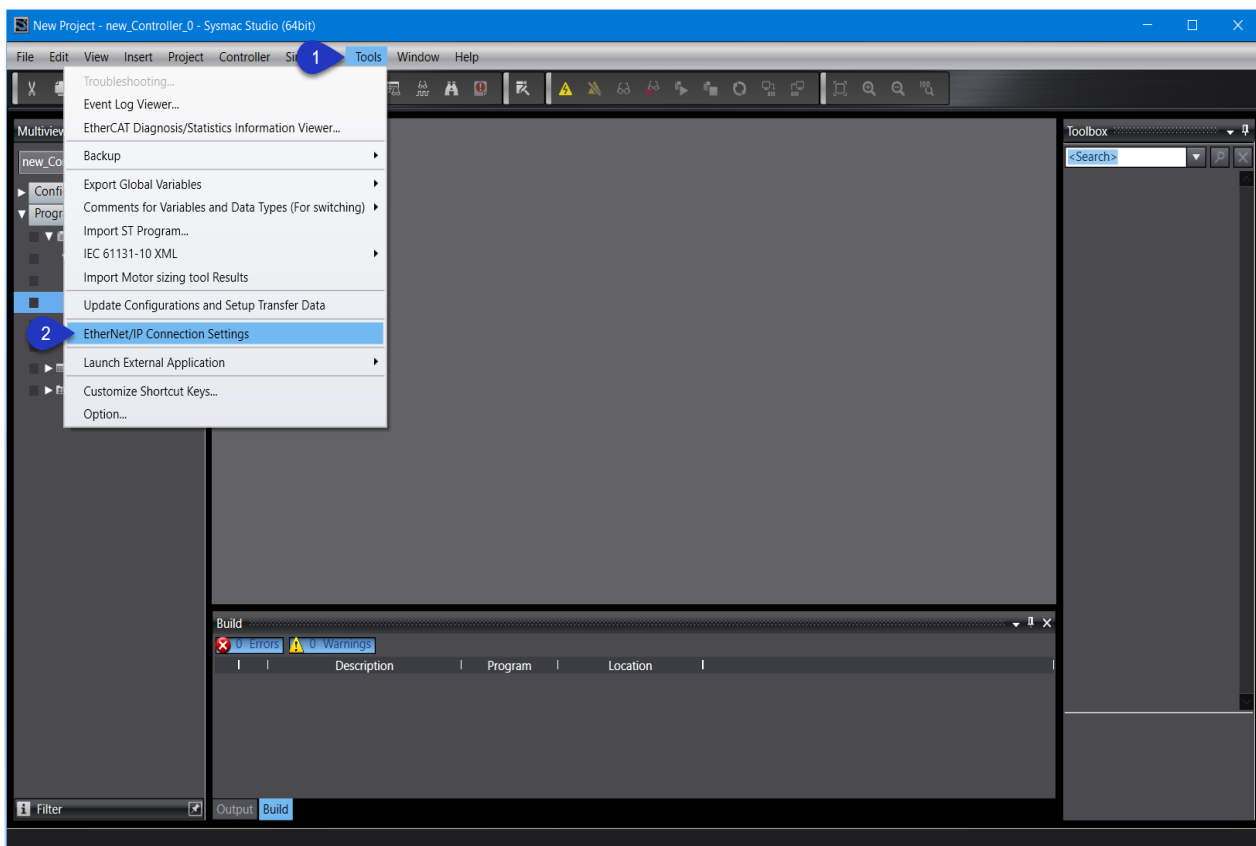
## C:\Program Files\OMRON\Sysmac Studio\IODeviceProfiles\EipConnection\Eds



### 4. Start the Sysmac Studio.

## Method 2

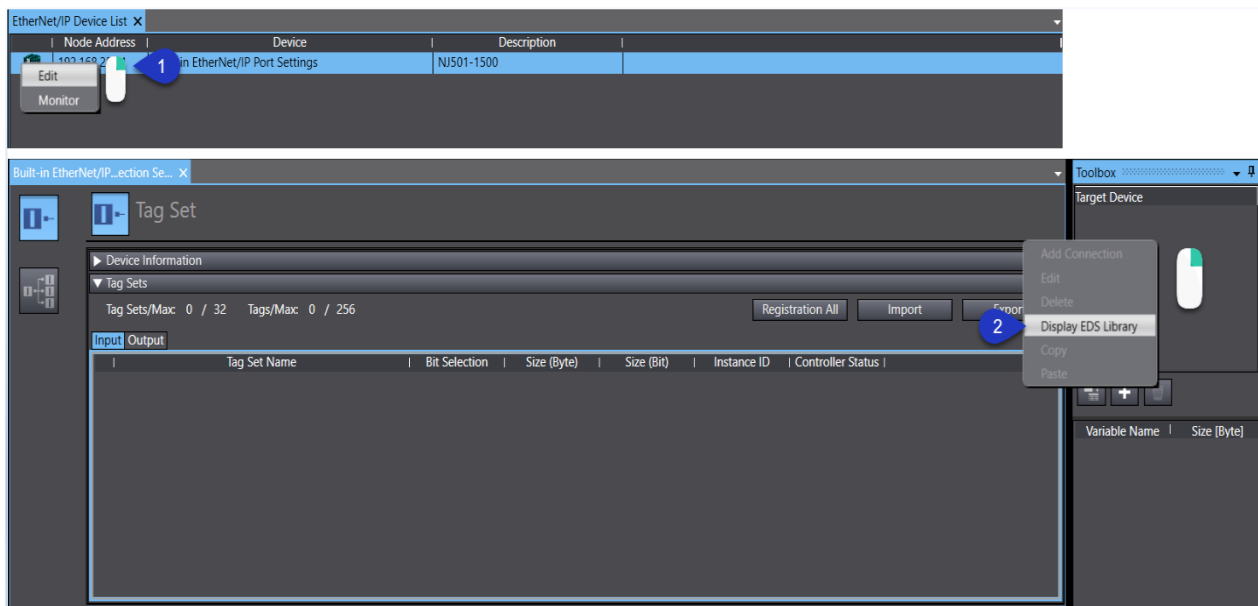
### Step 1



1. Open any Sysmac Studio Project and select “Tools” option from the toolbar.

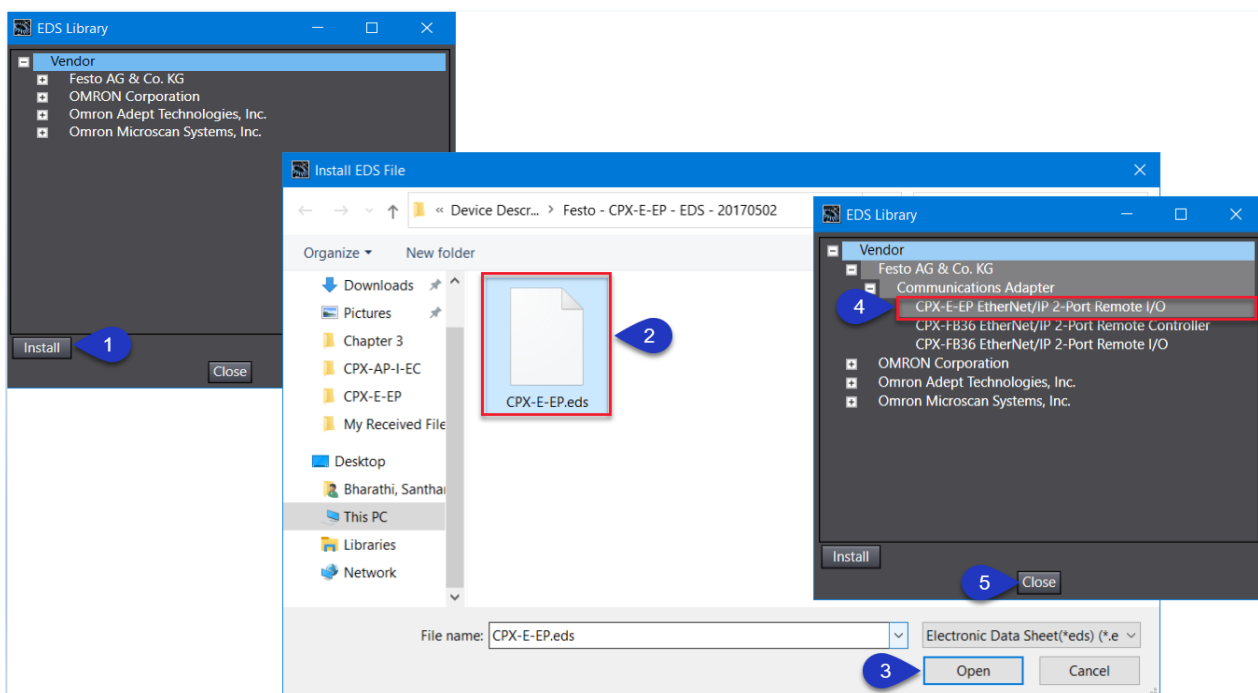
2. Select **"EtherNet/IP Connection Setting"** from tools menu.

## Step 2



1. Right click **"EtherNet/IP Device List"** shown on list and select **"Edit"** option.
2. In **"EtherNet/IP Port Setting Connection"** window Right click on **"Target Device"** area on toolbox then select **"Display EDS Library"** from menu.

## Step 3



1. Select **"Install"** option from **"EDS Library"** window.

2. Open EDS file location path on PC then select the required EDS file.
3. Select **“Open”** button.
4. Make sure installed EDS file available on **“EDS Library”** window.
5. Select **“Close”** button.

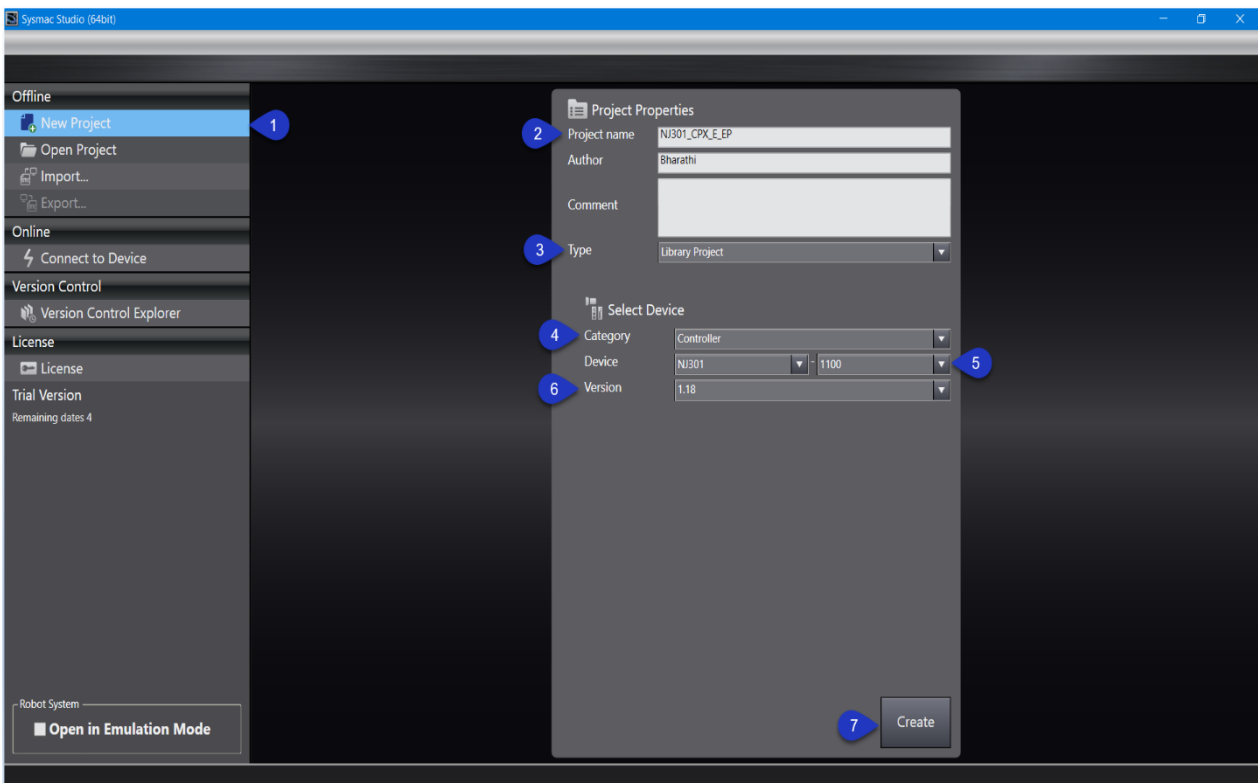
## 5.2 Starting a Project

Create a New Project selecting **Device Type and Version** that you have.



### Note

- The Device Version **MUST BE 1.10 OR HIGHER**. An older controller will not be compatible with the Festo EtherNet/IP library.
- Skip this chapter if user when integrating on existing projects.



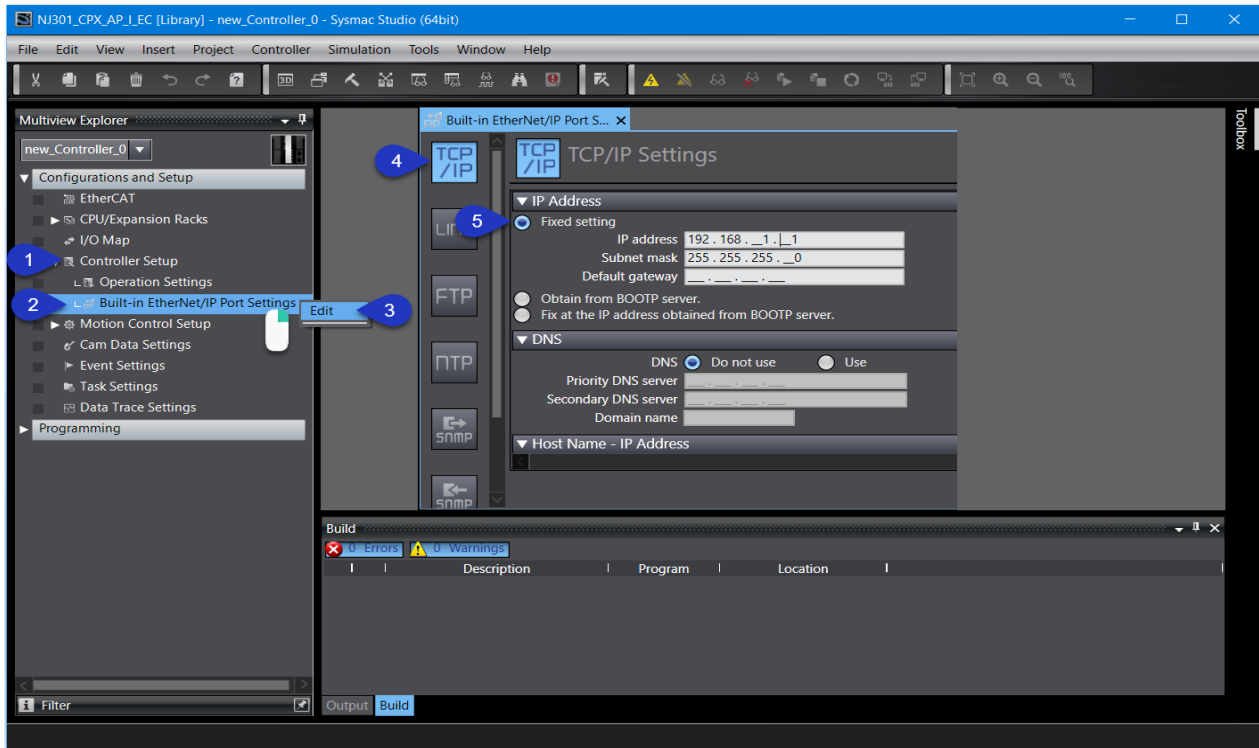
1. Open **“Sysmac Studio”** software and select **“New Project”** from the options.
2. Rename the **“Project Name”** from **“Project Properties”** window.
3. Select the **“Library Project”** from **“Type”** option.
4. Select **“Controller”** option in drop down menu of **“Category”** in select device.
5. Select the Device hardware you have from **“Device”** option in select device ( this sample code uses NJ301-1100 PLC ).
6. Select the device hardware version user have from **“Version”** option in select device ( this sample code uses Ver. 1.18 ).
7. Click **“Create”** button to create a new project.

## 5.3 PLC EtherNet/IP Address Configuration

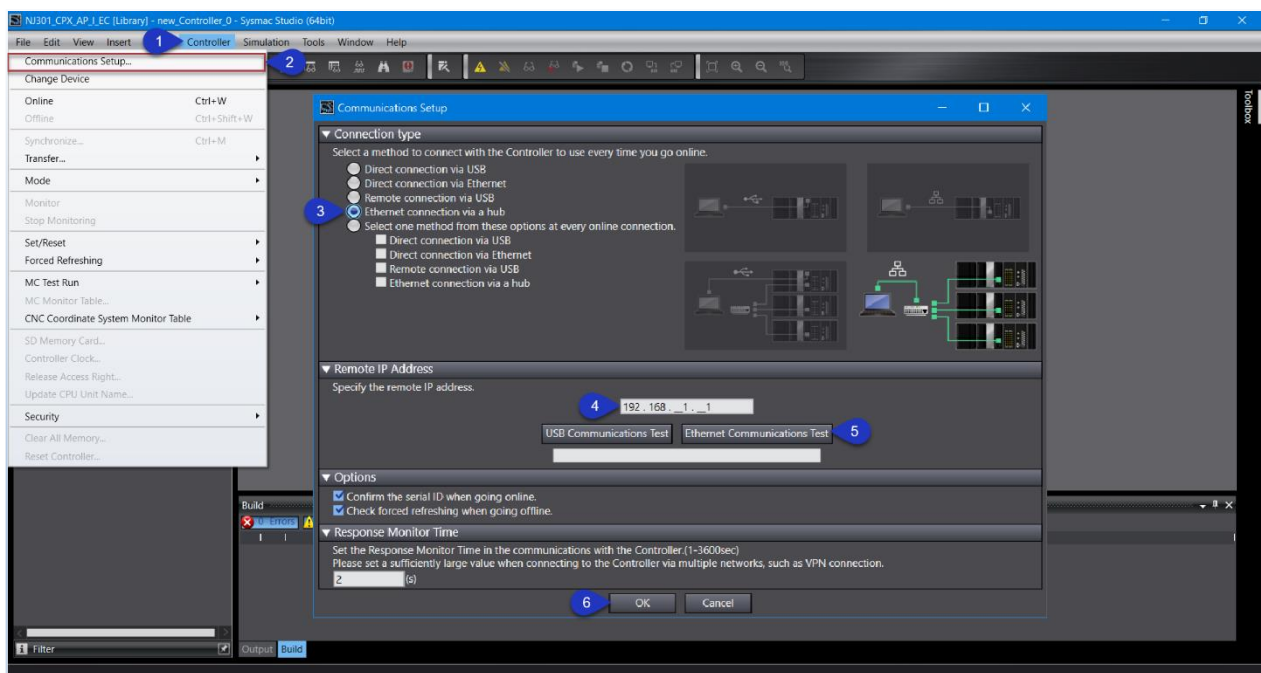


### Note

- Skip this chapter if user when integrating on existing projects.

**Step 1**

1. Expand **“Controller Setup”** setting.
2. Select **“Built-in EtherNet/IP Port Setting”** option.
3. Double click the **“Built-in EtherNet/IP Port Setting”** for open **“EtherNet/IP Settings”** window.
4. Select **“TCP/IP Settings”** tab to configure IP address.
5. Set the desire IP address and Subnet mask in **“Fixed Setting”** option.

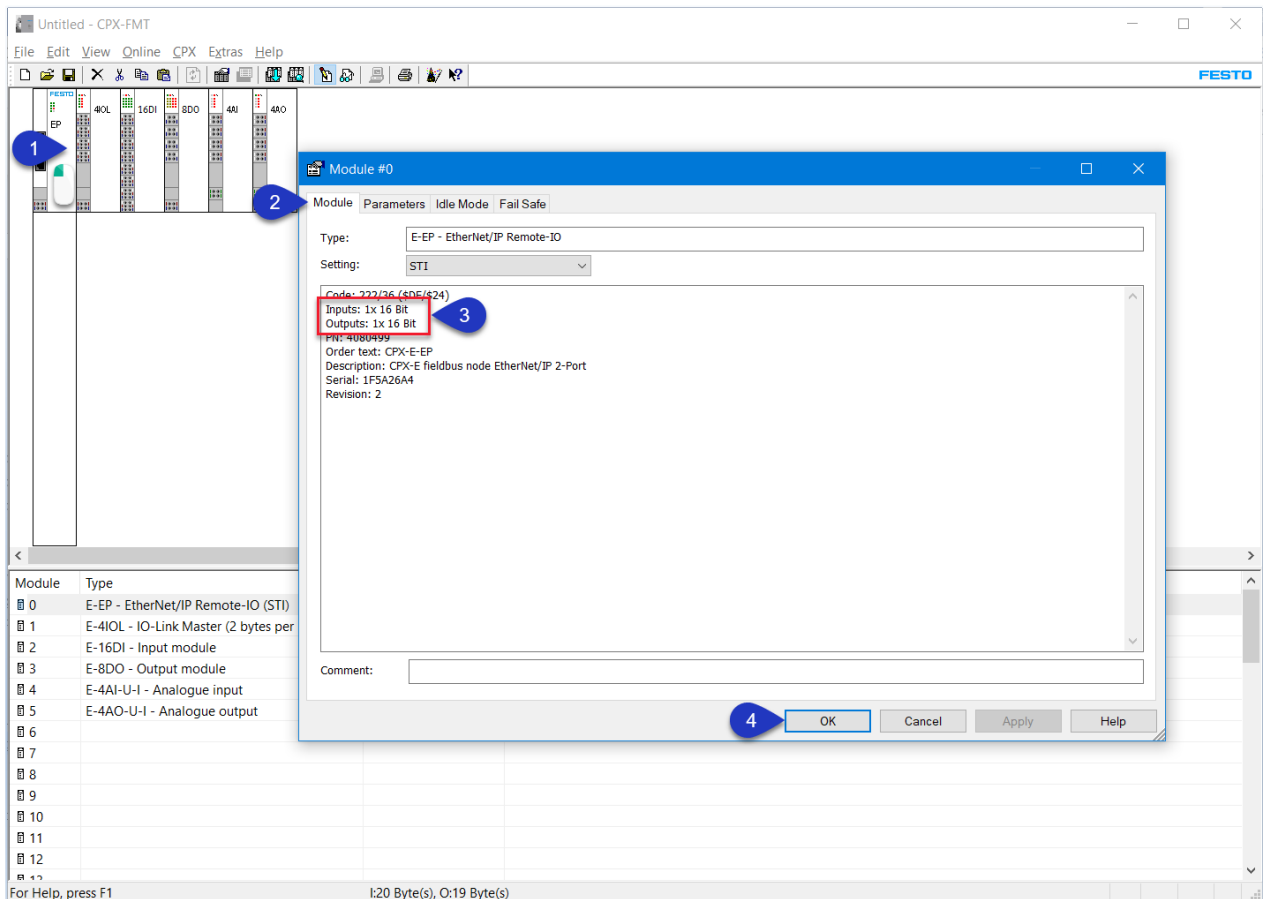
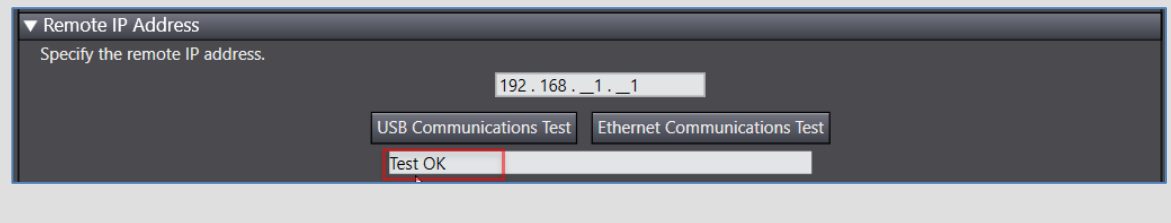
**Step 2**

1. Select **“Controller”** from the toolbar.
2. Select **“Communication Setup”** from controller menu.
3. Select **“Ethernet Connection”** from **“Connection Type”** as per required connection type.
4. Enter the IP address defined in step – 1.
5. Click **“Ethernet Communication Test”** button.
6. Click **“OK”** button.



**Note**

A successful test will display the message Test OK.



## 5.4 EtherNet/IP Target Device Variable Configuration

### Step 1

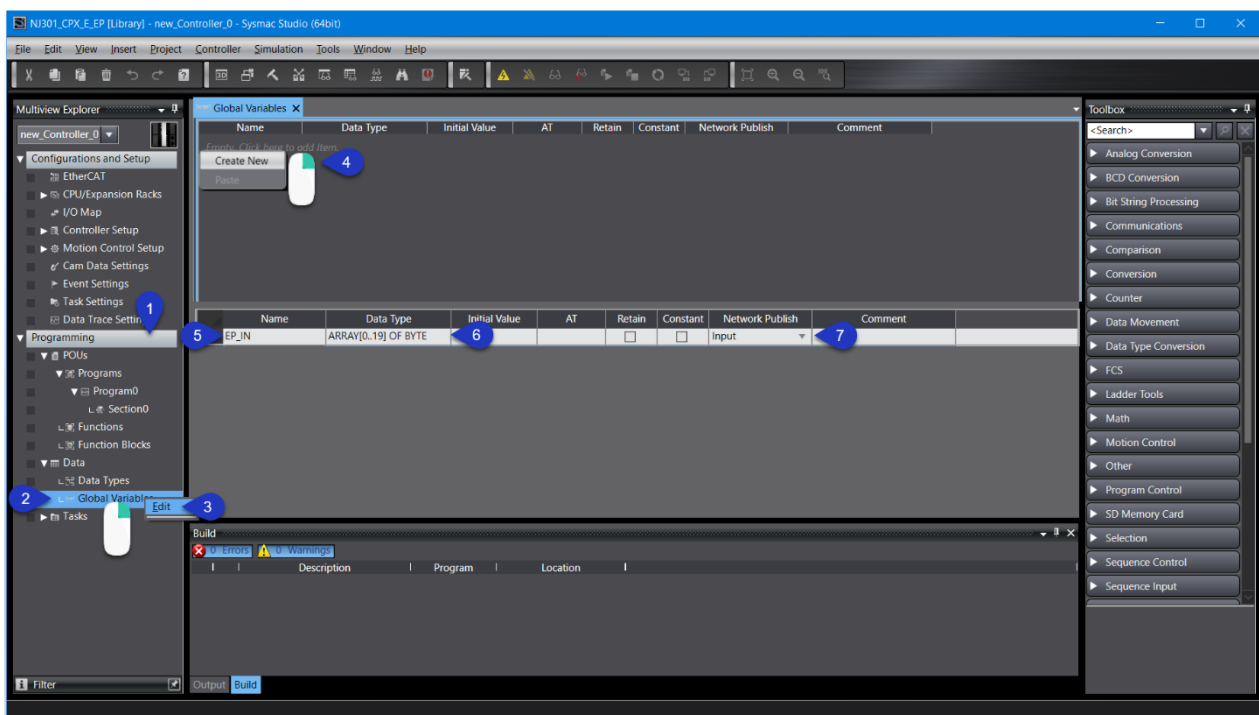
1. In CPX FMT software select EtherNet/IP module and double click to open module configuration window.
2. Select **“Module”** tab from module configuration window.
3. Check the input and output size highlighted in red box and convert as follows
  - a. 16 bit ---> 2 bytes
  - b. 32 bit ---> 4 bytes
4. Click **“OK”** button.

**Note**

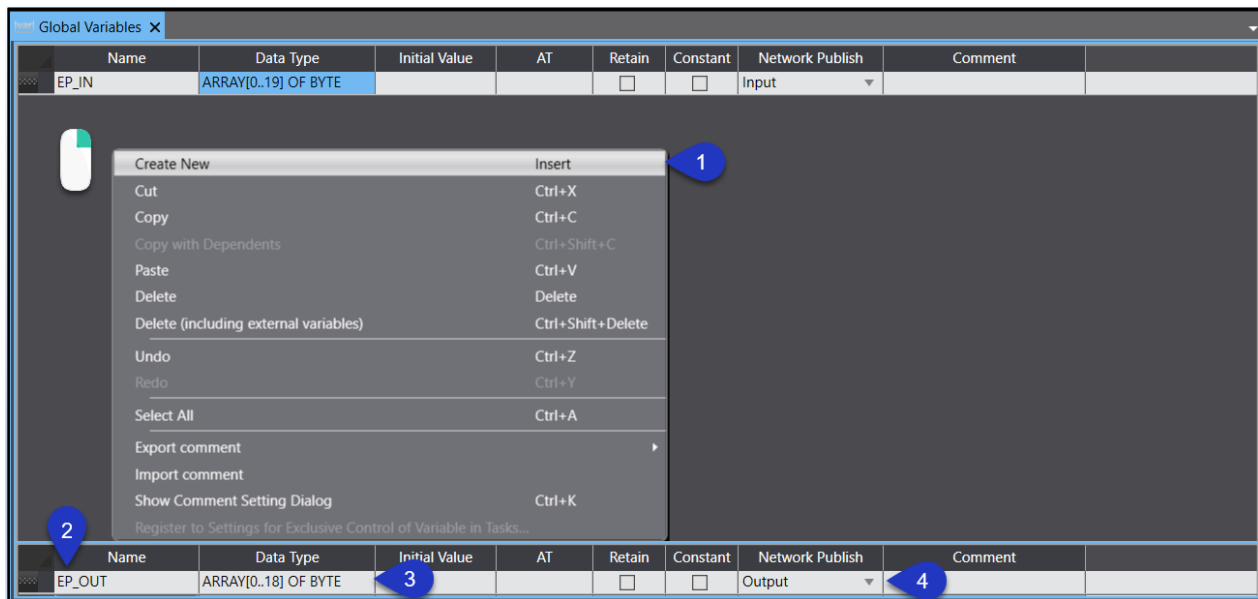
Repeat the same for Expansion Modules to calculate the IO size. As per this sample code use IO size as follows.

- EtherNet/IP Module            --->    2 bytes Input and 2 bytes Output
- IO-Link Module                --->    8 bytes Input and 8 bytes Output
- Digital Input Module           --->    2 bytes Input
- Digital Output Module         --->    1 bytes Output
- Analog Input Module          --->    8 bytes Input
- Analog Output Module         --->    8 bytes Output

**Total IO size => 20 bytes Input and 19 bytes Output**

**Step 2**

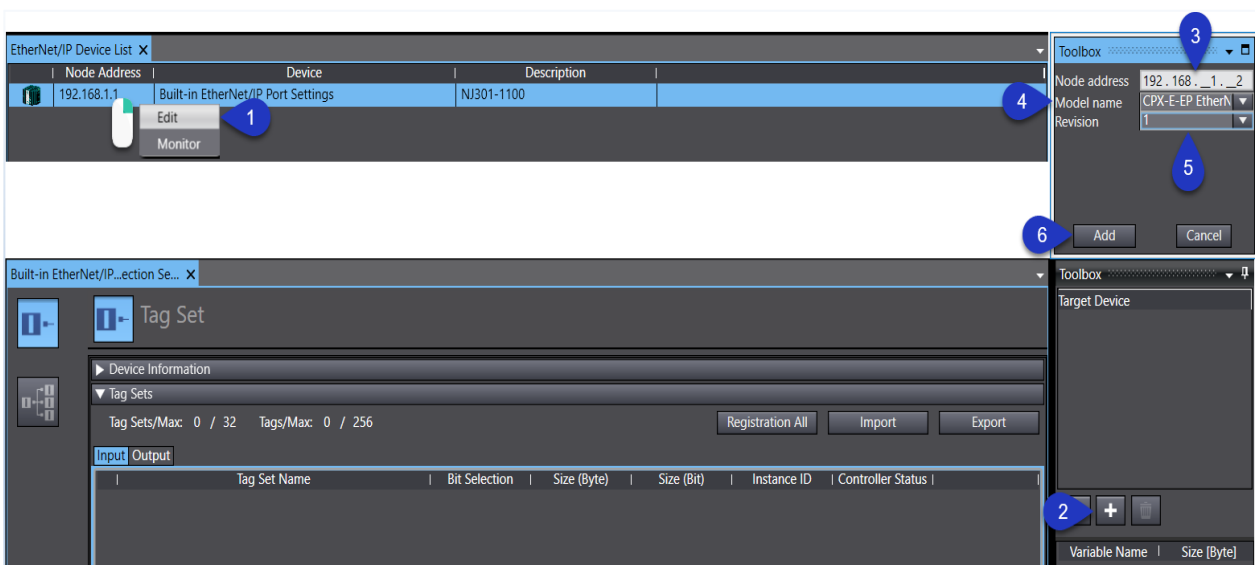
1. In Sysmac Studio software expand **“Programming”** header in Multiview Explorer.
2. Expand **“Data”** item from programming header.
3. Double click the **“Global Variables”** item to open variable edit pane.
4. Right click at variable edit pane and select **“Create New”** option.
5. Enter desired input variable name in **“Name”** box.
6. Make data type as **“Array of Bytes”** in **“Data Type”** box (Size : 20 Bytes ).
7. Select **“Input”** option from drop down menu in **“Network Publish”** box.

**Step 3**

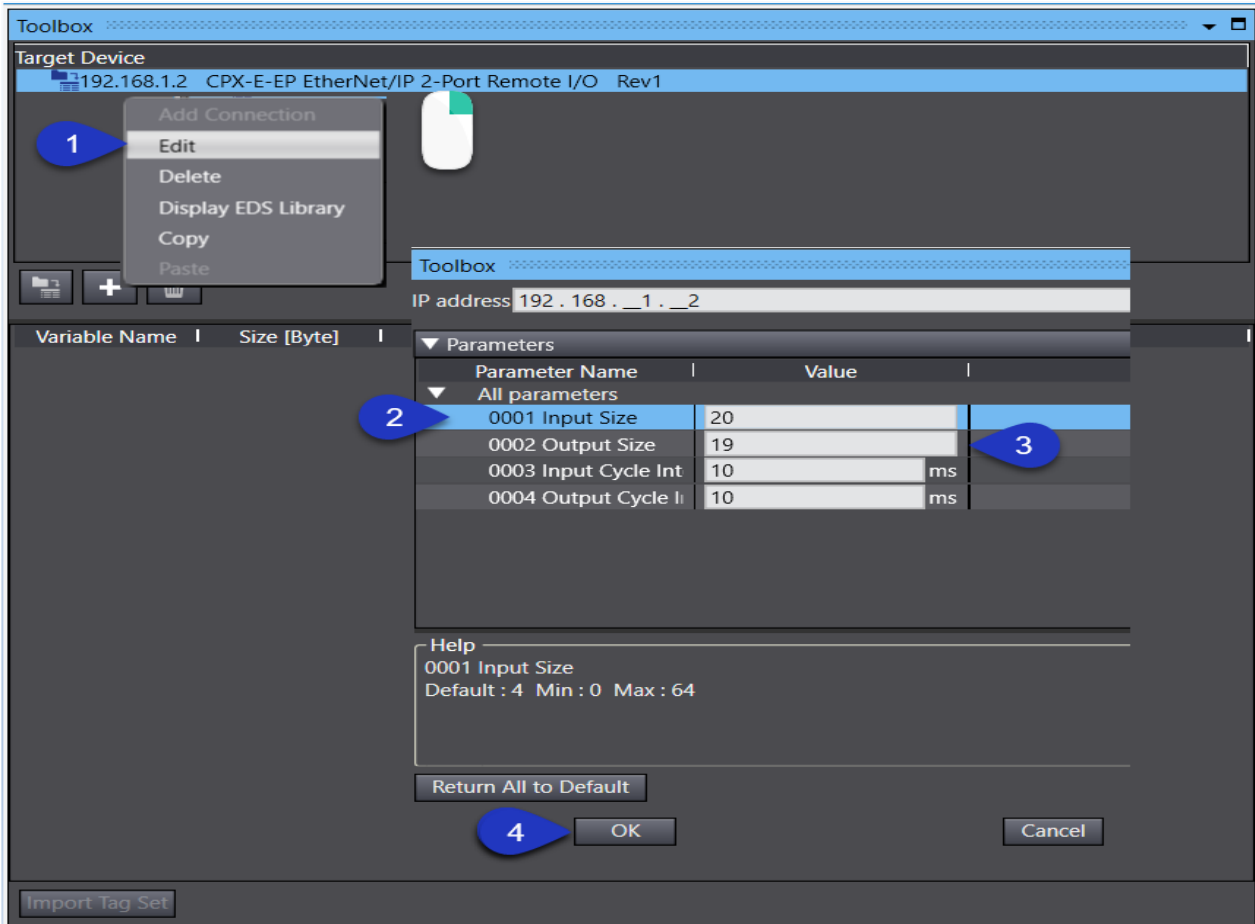
1. Right click at variable edit pane again and select **“Create New”** option.
2. Enter desired output variable name in **“Name”** box.
3. Make data type as **“Array of Bytes”** in **“Data Type”** box (Size : 19 Bytes ).
4. Select **“Output”** option from drop down menu in **“Network Publish”** box.

**5.5 EtherNet/IP Target Device Configuration****5.5.1 Adding the Target Device****Step 1**

To go to **“Built-in EtherNet/IP Port Setting”** for target device configuration follow the chapter 4.1 step – 2 method – 2 step – 1. If **“Built-in EtherNet/IP Port Setting”** is opened jump to step – 2.

**Step 2**

1. Right click **"EtherNet/IP Device List"** shown on list and select **"Edit"** option.
2. Click **+** button from **"Toolbox"** window to add **"Target Device"** for ethernet/ip network **"EtherNet/IP Port Setting Connection"** screen.
3. Enter the node address of **"Target Device"** on toolbox window.
4. Select the model name of **"Target Device"** from drop down list on toolbox window.
5. Select the revision of **"Target Device"** from drop down list on toolbox window.



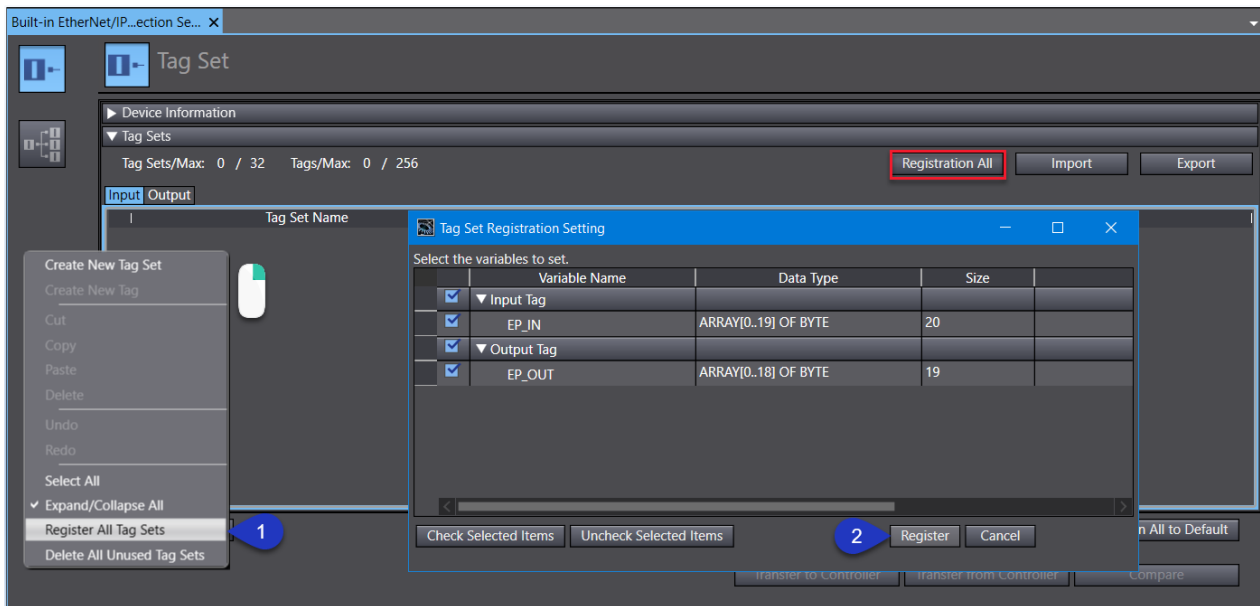
6. Click **"Add"** button to add the **"Target Device"** to the network.

### Step 3

1. Right click the target device which added in step – 2 in toolbox window and select **"Edit"**.
2. Enter the input size in **"Target Device Parameters"** in toolbox window as described in Chapter 4.4 ( Input : 20 Bytes ).
3. Enter the output size in **"Target Device Parameters"** in toolbox window as described in Chapter 4.4 ( Output : 19 Bytes ).
4. Click **"OK"** button.

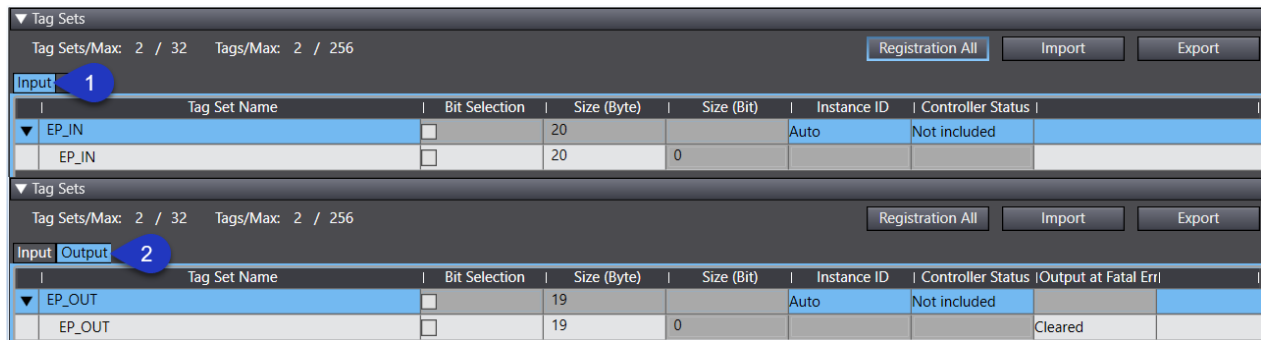
## 5.5.2 Register the Tag Set for Target Device

### Step 1



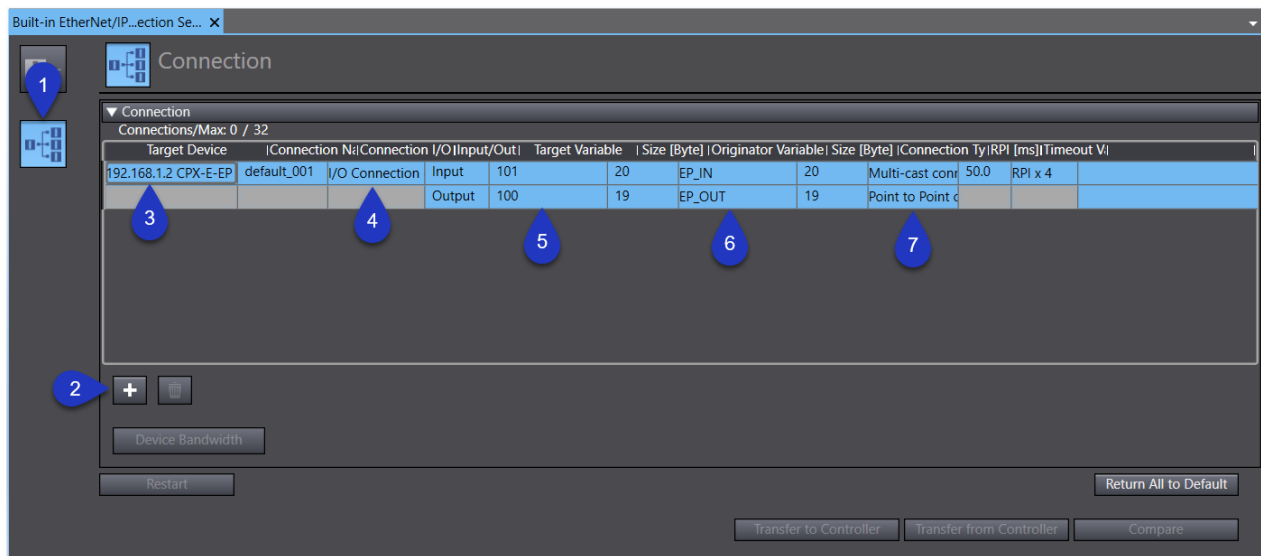
1. Right click on “Tag Set” area at “EtherNet/IP Port Setting Connection” pane and select “Register All Tag Set” or simply click “Register All” button as highlighted in red box.
2. Check all tag set size with target device parameters at “Tag Set Registration Setting” window then click “Register” button.

### Step 2



1. Registered input tag set with “Size” and “Instance ID” as shown on tag set area.
2. Registered output tag set with “Size” and “Instance ID” as shown on tag set area.

### 5.5.3 Target Device Input / Output Connection



1. Click **“Connection”** button on **“EtherNet/IP Port Setting Connection”** pane.
2. Click **“+”** button to add the target device at connection.
3. Select the **“Target Device”** from drop down menu.
4. Select the **“Connection IO”** format from drop down menu.
5. Enter the **“Target Variable”** for IO ( Standard Value for Input => 101, Output => 100 ).
6. Select the **“Originator Variable”** from drop down menu for IO.
7. Select the **“Connection Type”** from drop down menu for IO.

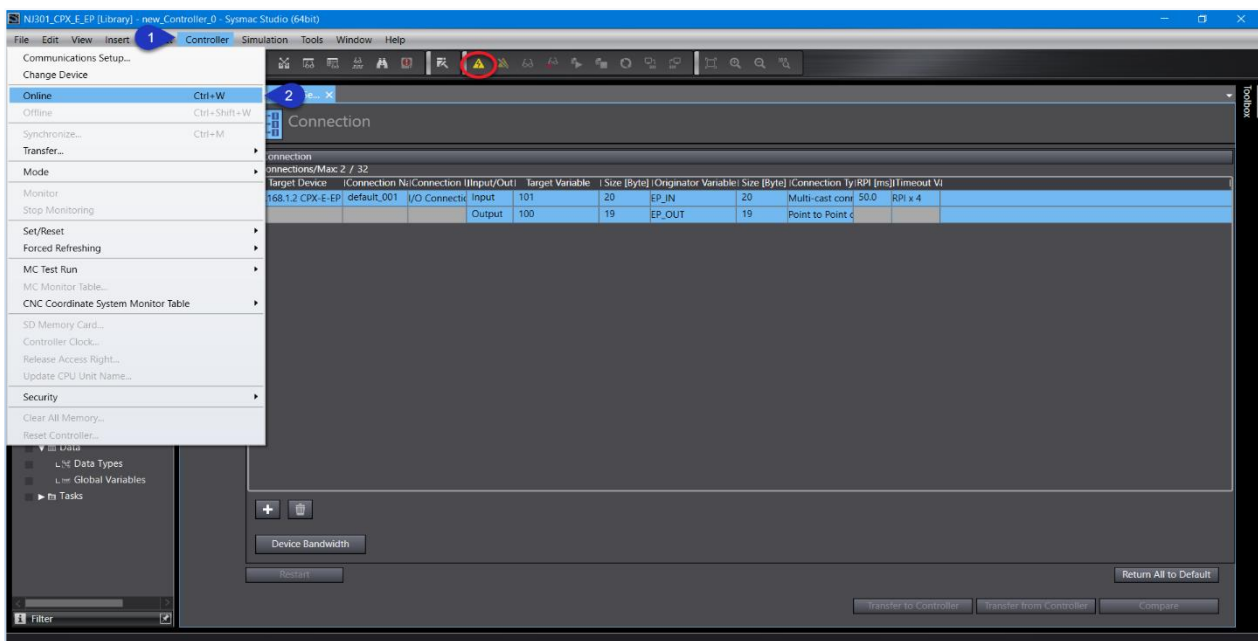
### 5.6 PLC Configuration Transfer to Controller



#### Warning

Before transfer make sure there is no physical activity.

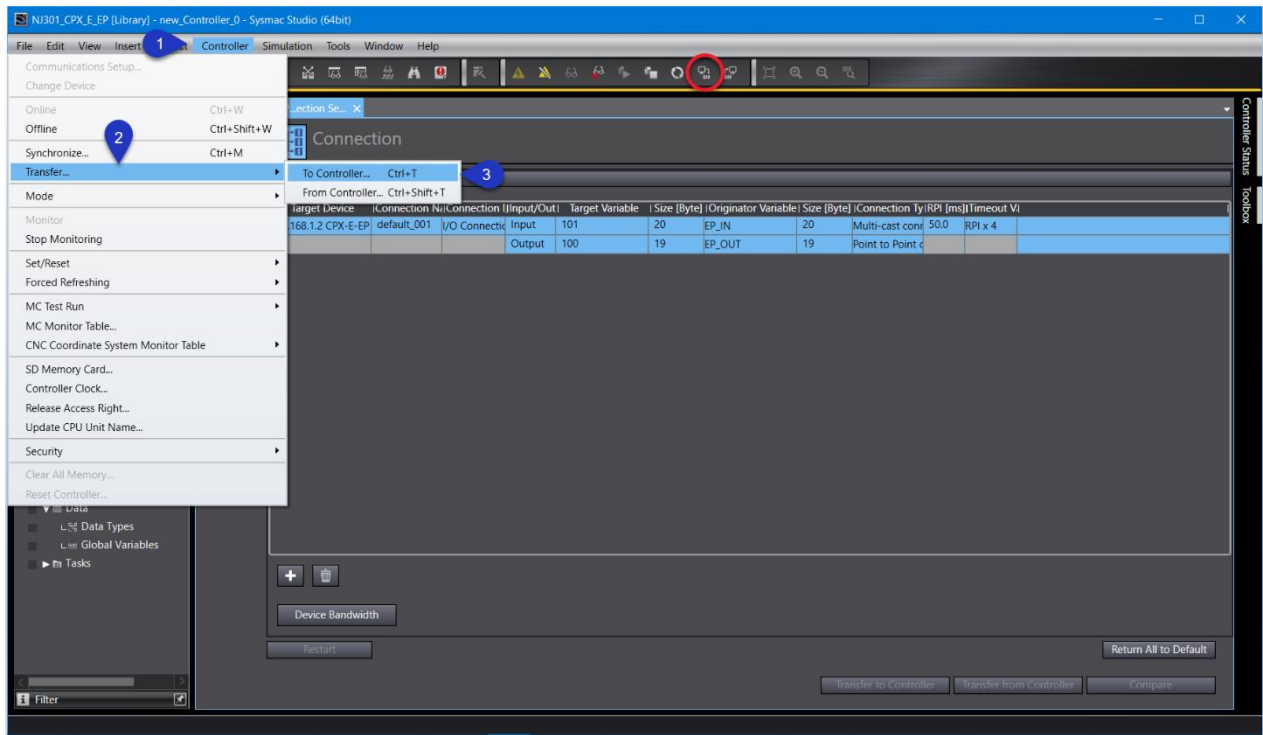
#### Step 1



1. Select **“Controller”** from the toolbar.

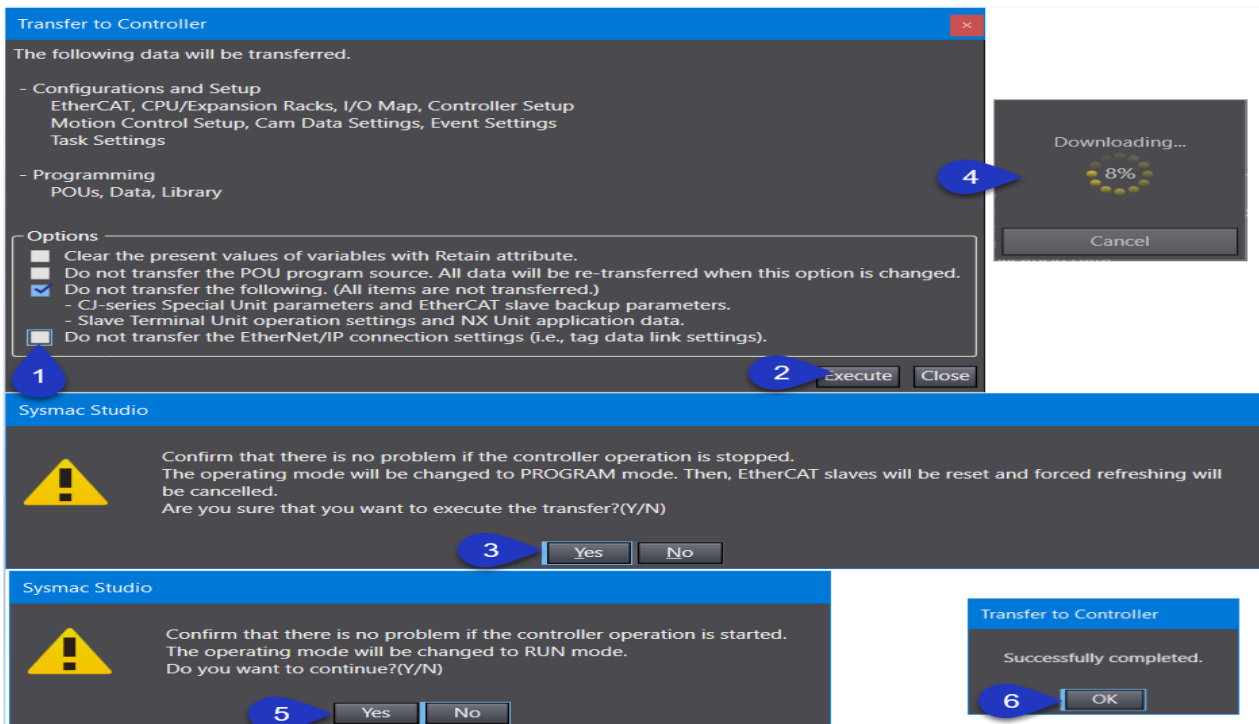
- Click **“Online”** from controller menu. User can also use either the shortcut button shown in red circle or press **“Ctrl+W”** shortcut key to go online.

## Step 2



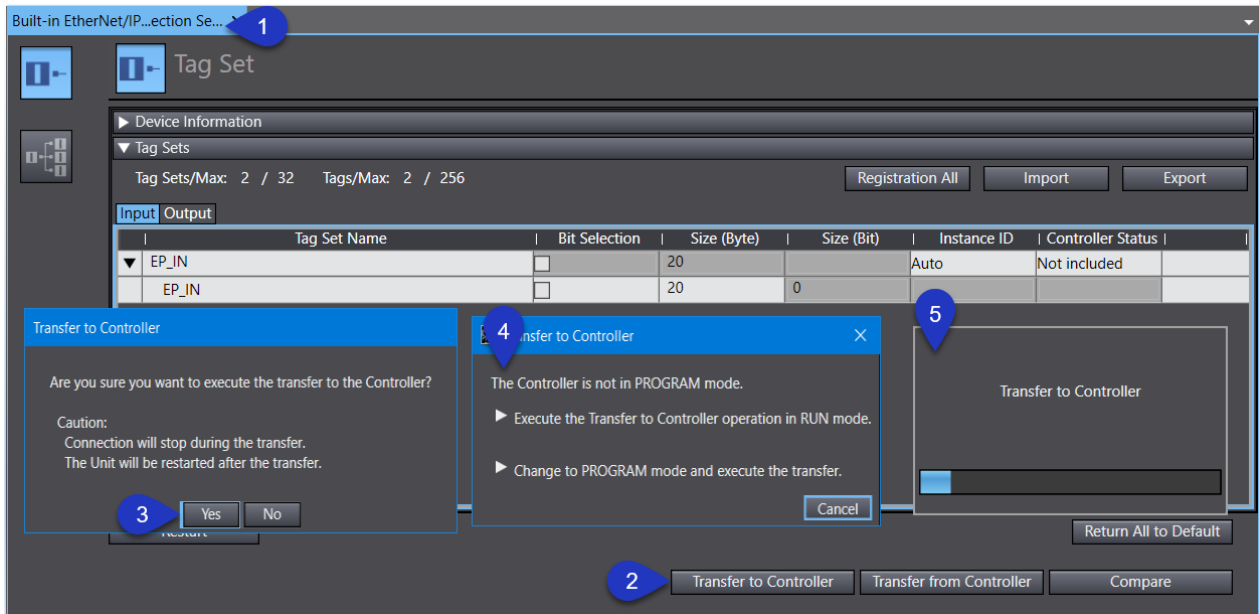
- Select **“Controller”** from the toolbar.
- Select **“Transfer”** from controller menu.
- Select **“To Controller”** from transfer option. User can also use either the shortcut button shown in red circle or press **“Ctrl+T”** shortcut key to transfer to controller.

## Step 3



1. Uncheck the **“Do not transfer the EtherNet/IP connection settings”** parameter.
2. Click **“Execute”** button from **“Transfer to Controller”** window as per selection of options as shown.
3. Click **“Yes”** to start transferring.
4. Downloading window shows project has been transferring to controller.
5. Click **“Yes”** to finish transferring.
6. Click **“OK”** button after **“Successfully Completed”** message display from transfer to controller window.

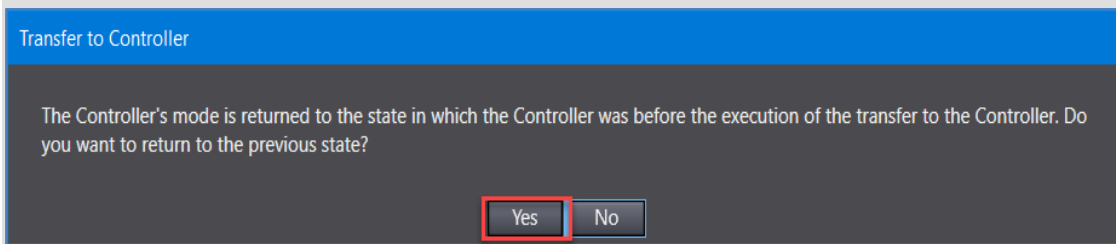
#### Step 4



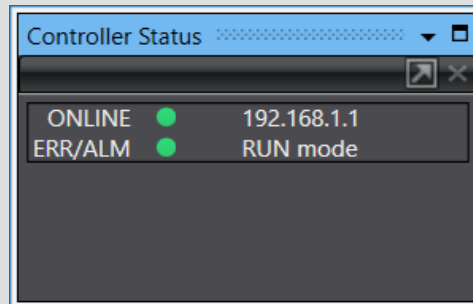
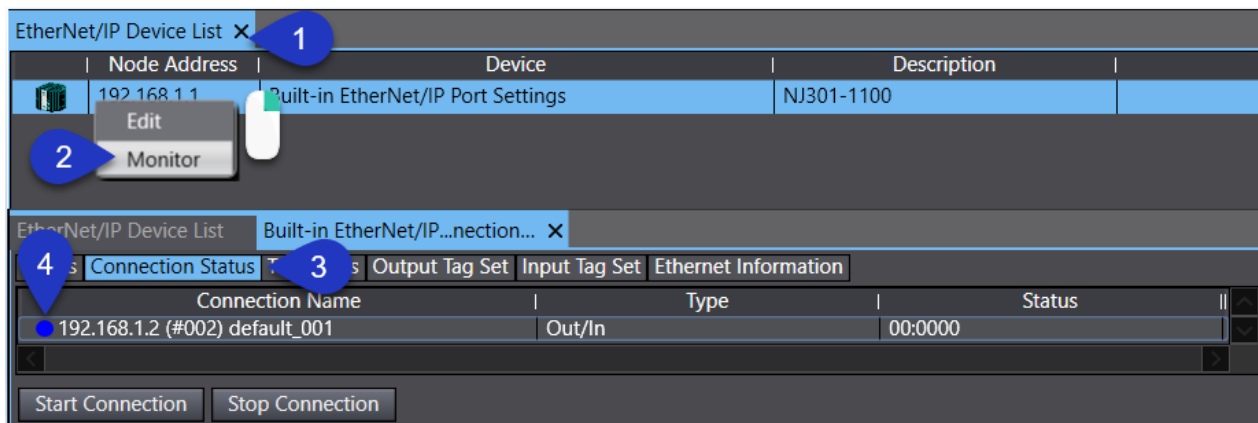
1. Go to **“EtherNet/IP Port Connection Setting”** as did in chapter 4.1 step – 2 method – 2 step – 1.
2. Click **“Transfer to Controller”** button.
3. Click **“Yes”** button in **“Transfer to Controller”** window.
4. Select either **“Method – 1”** or **“Method – 2”** as per user application condition ( If Controller is in Run mode then only this window will pop-up ).
5. After successful completion **“Transfer to Controller”** window automatically closed.

**Note**

If you select **“Method – 2”** then you have to click **“Yes”** button after successful completion.



After successful completion check controller status.

**Step 5**

1. Go to **“EtherNet/IP Device List”** as did in chapter 4.1 step – 2 method – 2 step – 1.
2. Right click **“EtherNet/IP Device List”** shown on list and select **“Monitor”** option.
3. Click **“Connection Status”** tab from **“EtherNet/IP Port Setting Connection Monitor”** window.
4. If EtherNet/IP connection is successfully integrated with target device then status indication blink in **“Blue”** colour. If there is any error status indication blink in **“Red”** colour.

## 6 Addressing Concept

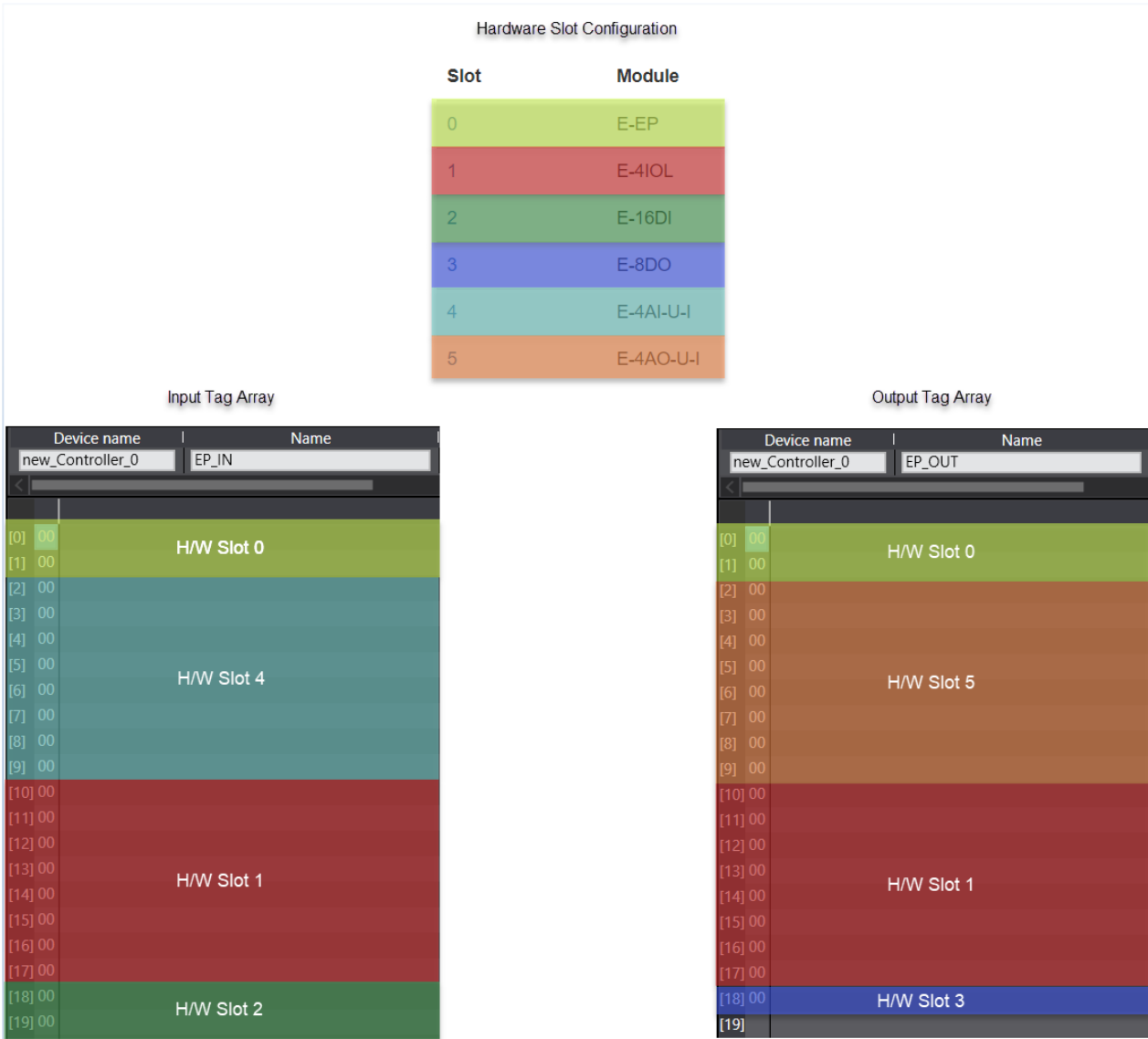
Address tags are classified into three types based on its functionality. They are input tags, Output tags & Configuration tags. Input tags receives feedback from field device, Output tags gives command to field devices & Configuration tags are used to set the configuration parameters of field devices. This topic will give an idea to address the Counter module in PLC program.



**Note**

- Addressing format of the modules in PLC program will not be as like slot position. It will get rear-ranged depending on the module hardware type. CPX-E-EP module diagnose tags have first priority in tags array, Technology modules are considered as second priority, Analog modules are considered as third priority, IO-Link modules are considered as fourth priority & Digital modules are considered as last priority in IO tag array position. Upcoming topics will give detailed explanation about addressing of Digital, Analog & IO-Link Modules.

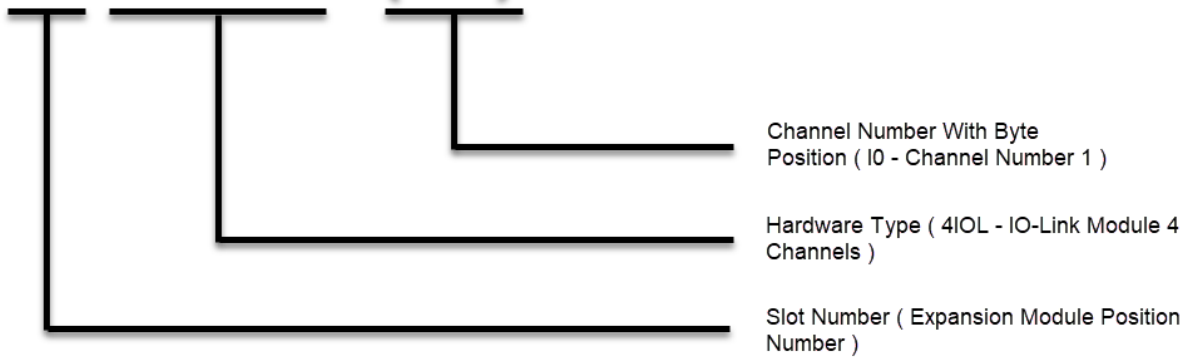
For an example consider user has the below hardware configuration. As per the HW configuration in the PLC Program Input & Output tags occupies address in array as shown below



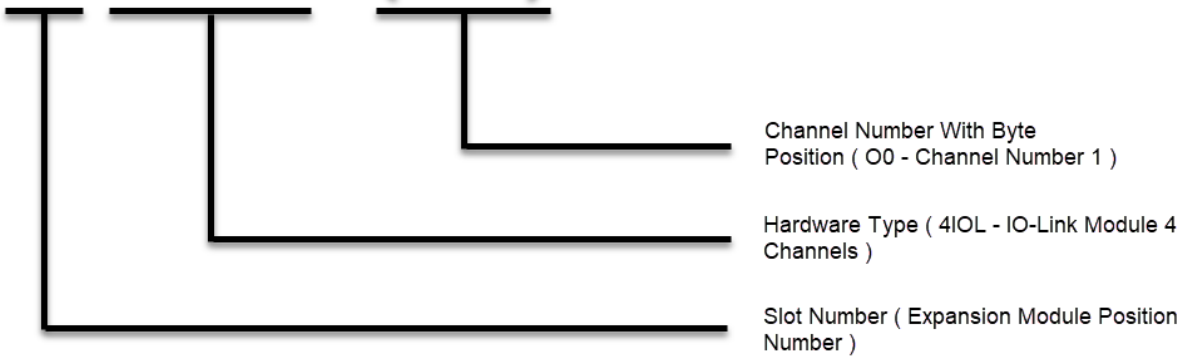
## 6.1 Addressing of IO-Link Modules

IO-Link channel address is split in to bytes. To find the correct bit address of input & output channels see the description of tag array. Below illustration gives idea to find bit addresses of input module's.

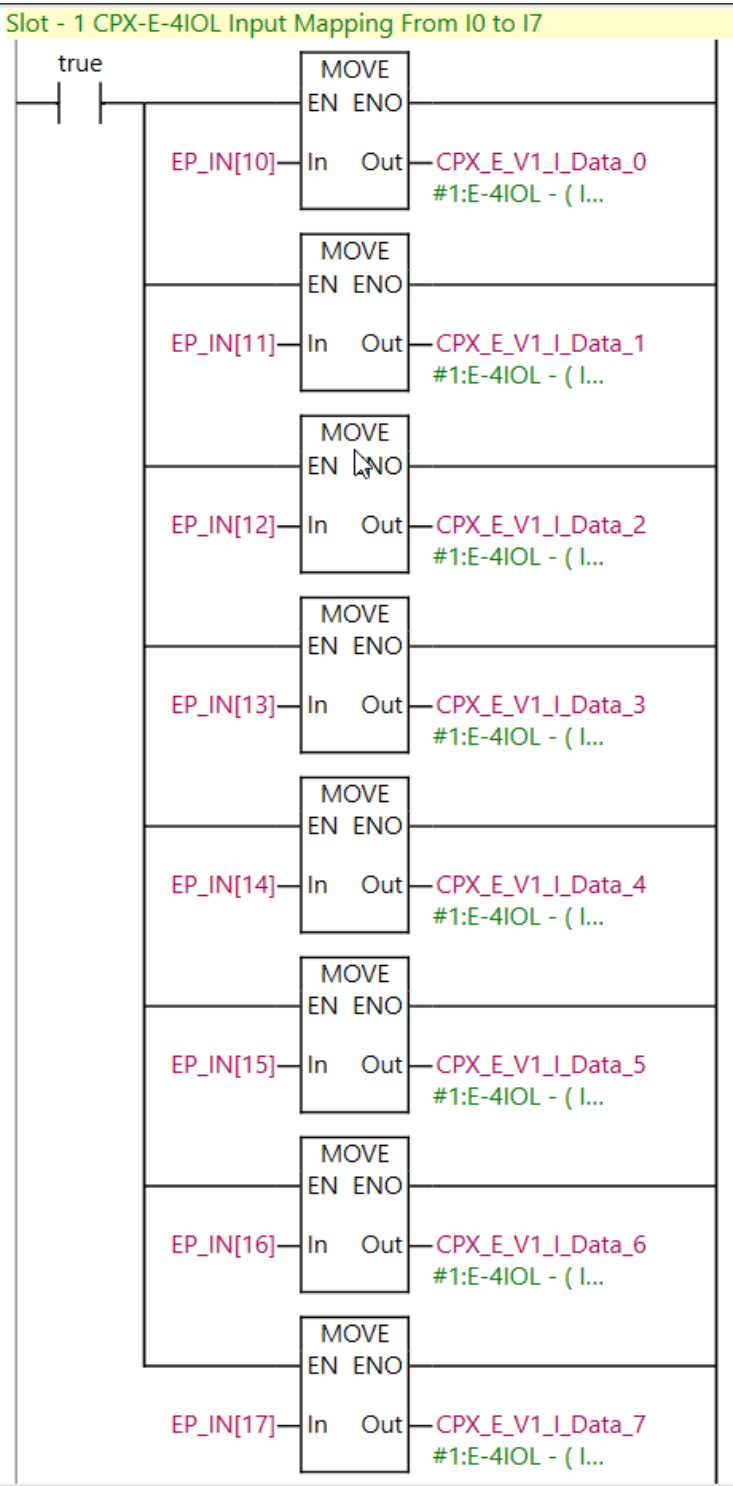
**#1:E-4IOL - ( I0 )**



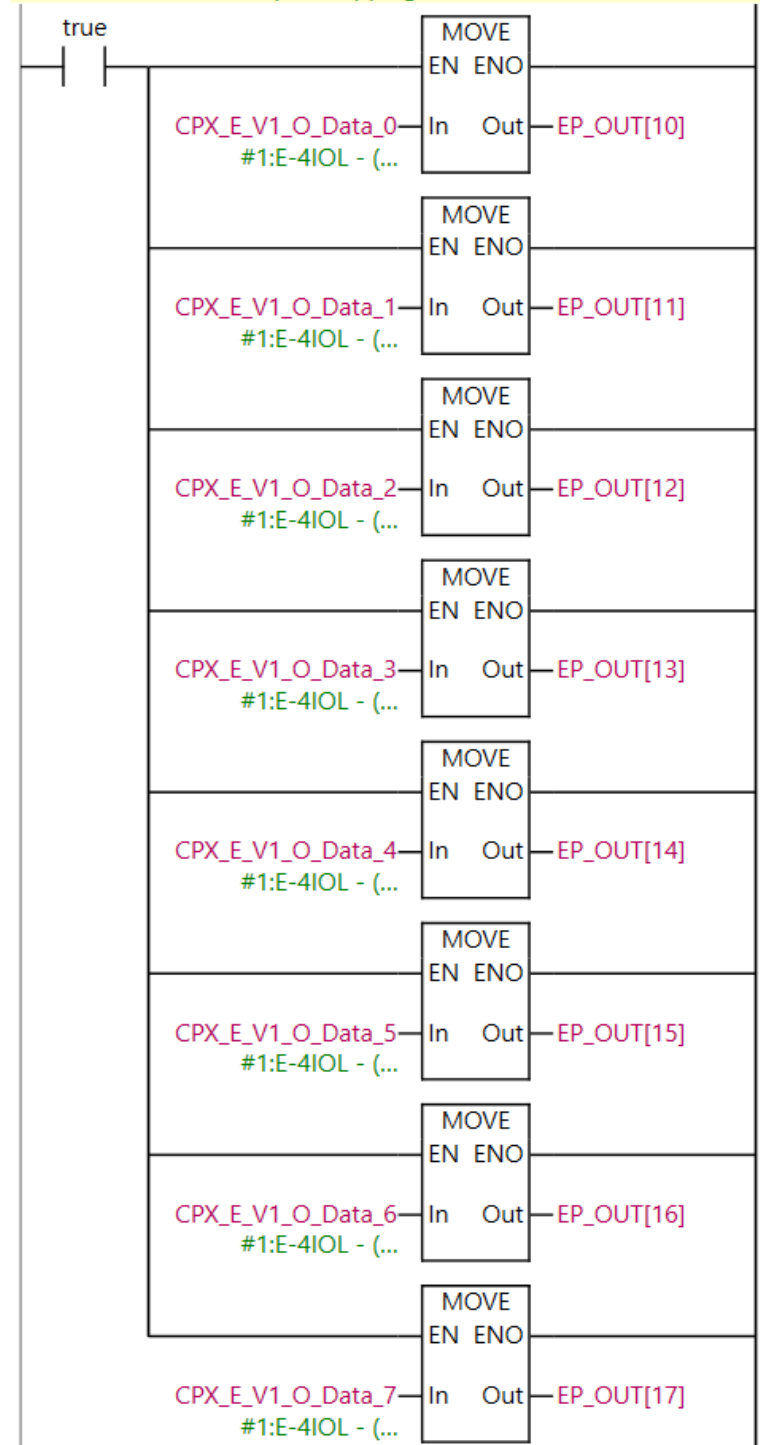
**#4:E-4IOL - ( O0 )**



6.2 Sample Code for IO-Link Module

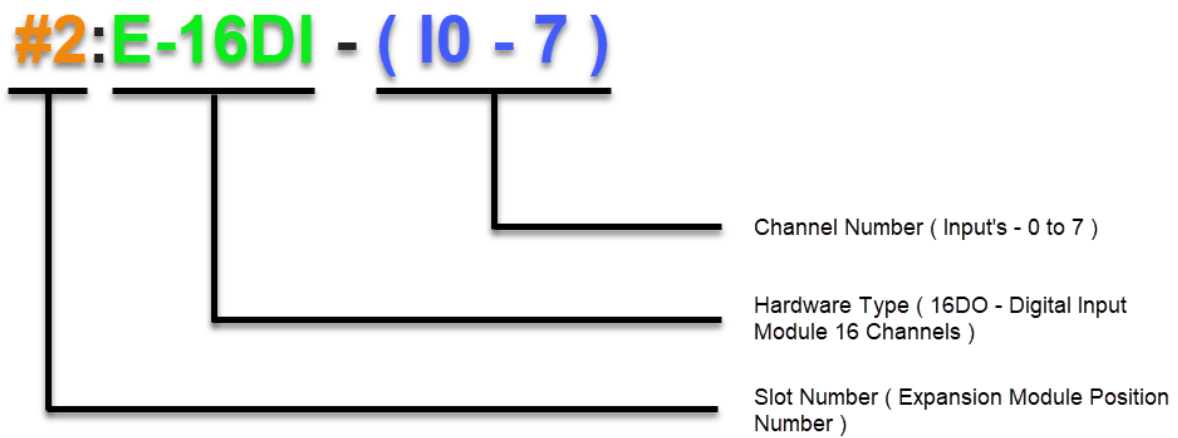


Slot - 1 CPX-E-4IOL Output Mapping From O0 to O7



### 6.3 Addressing of Digital Input Modules

To find the correct bit address of input channels see the description of input tag array & to find the correct bit address of input channels see the description of input tag array. Below illustration gives idea to find bit addresses of input module's.



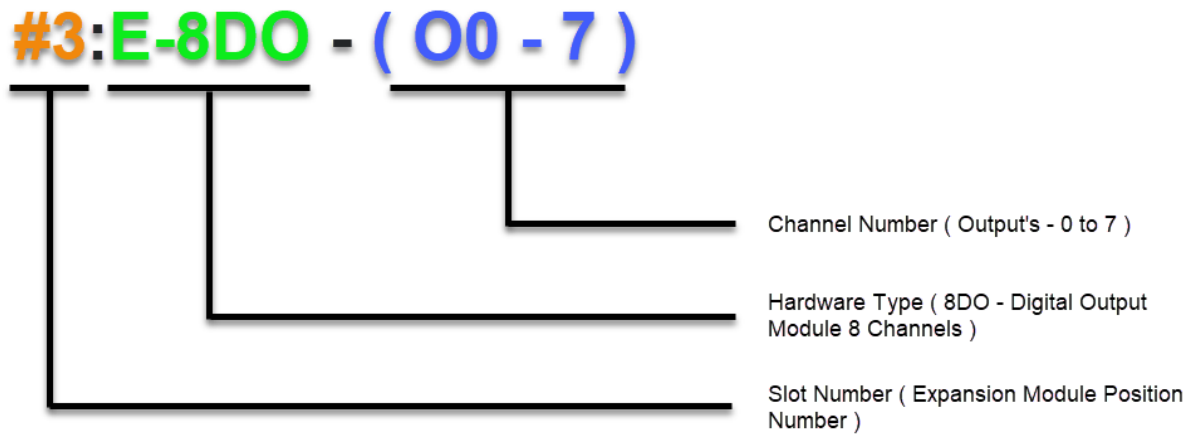
6.4 Sample Code for Digital Input Module



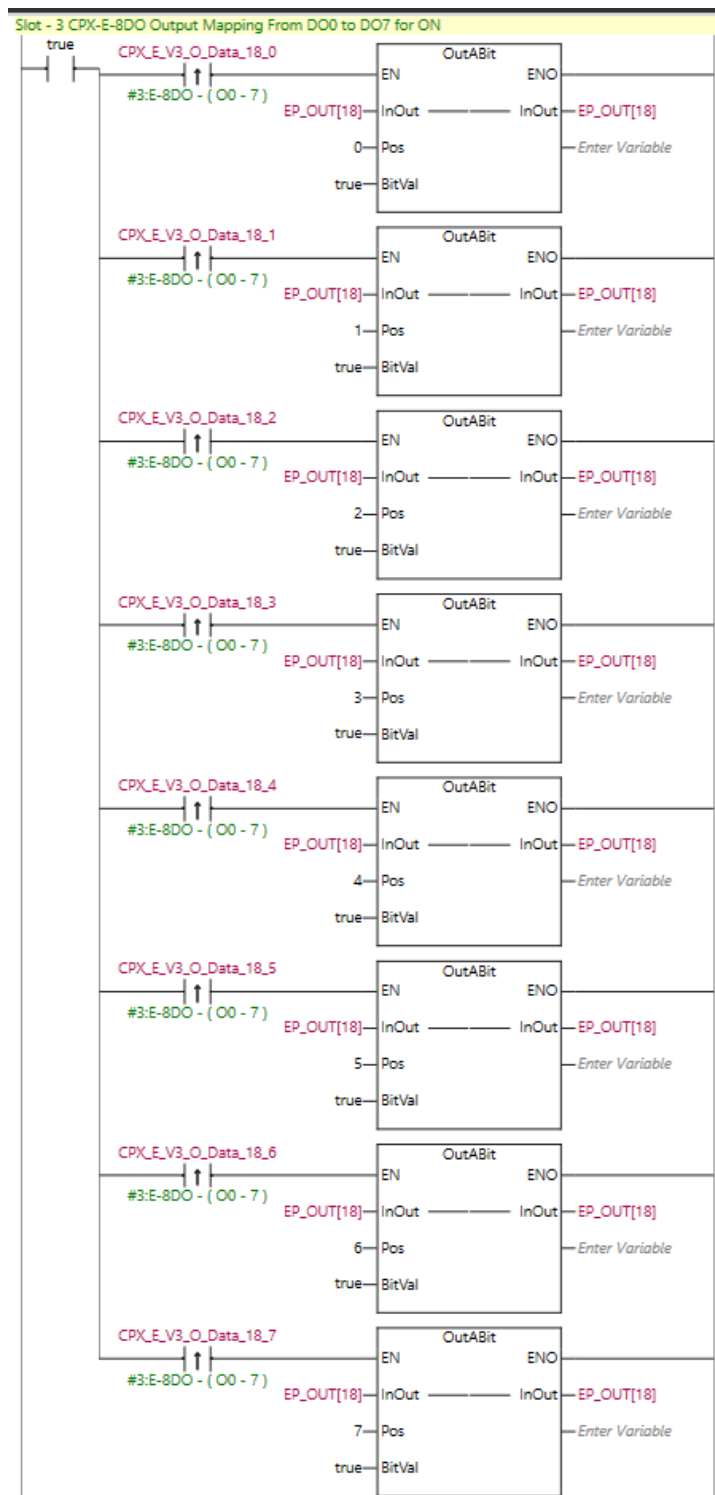


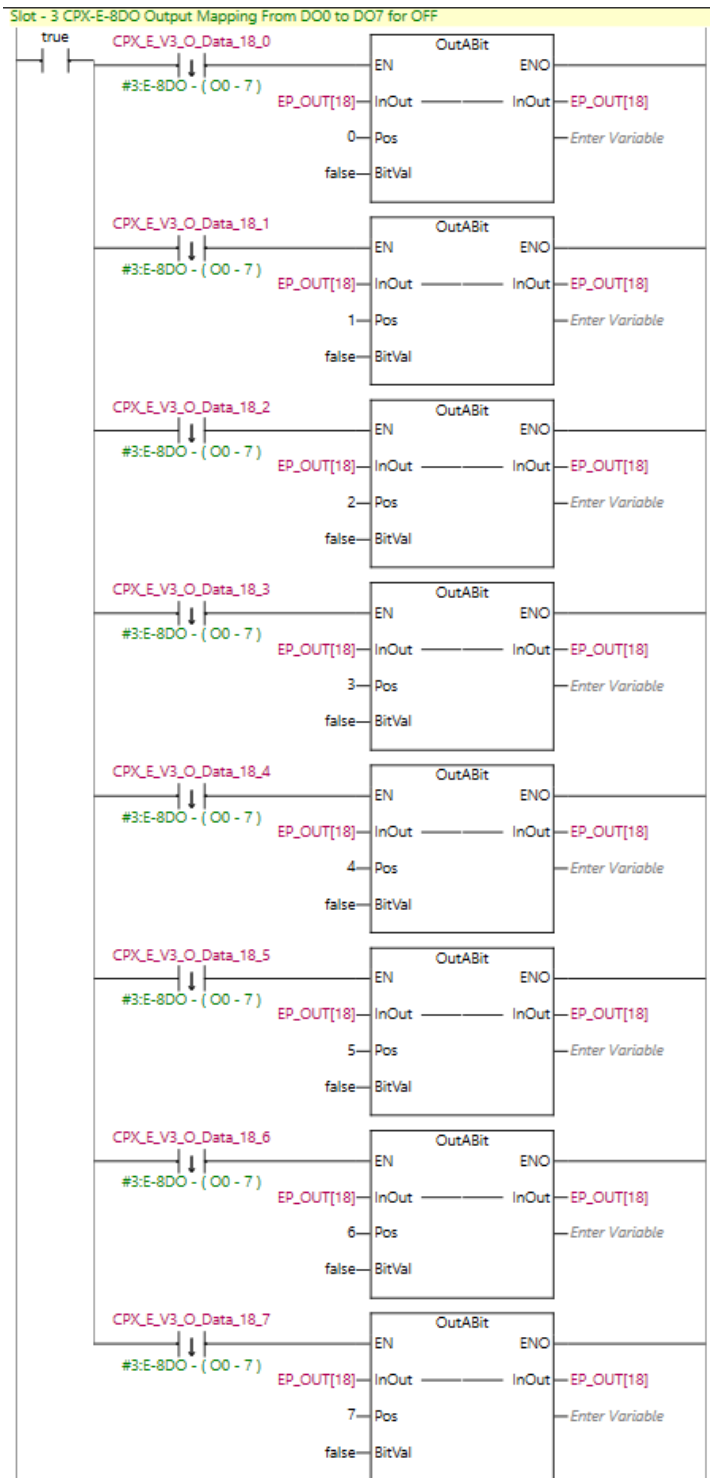
## 6.5 Addressing of Digital Output Modules

To find the correct bit address of output channels see the description of output tag array & to find the correct bit address of output channels see the description of output tag array. Below illustration gives idea to find bit addresses of output module's.



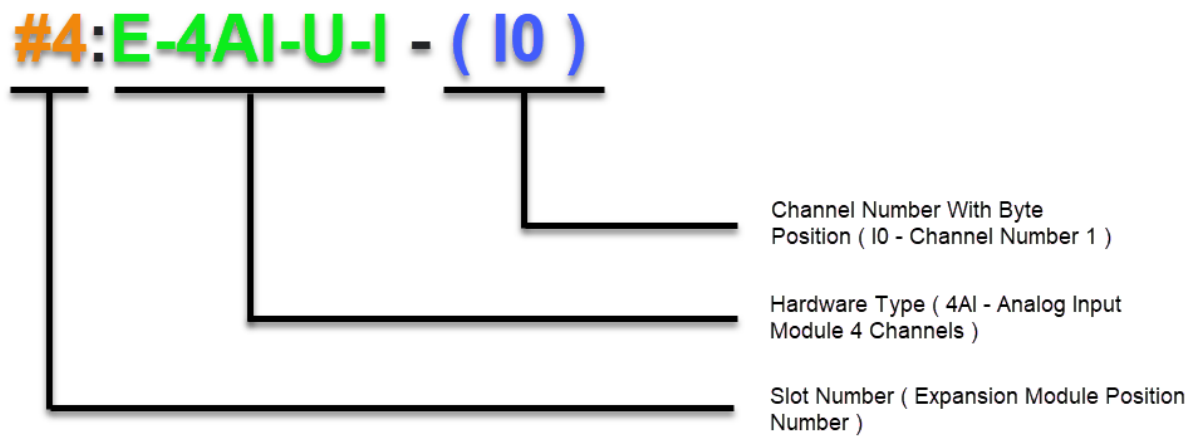
## 6.6 Sample Code for Digital Output Module





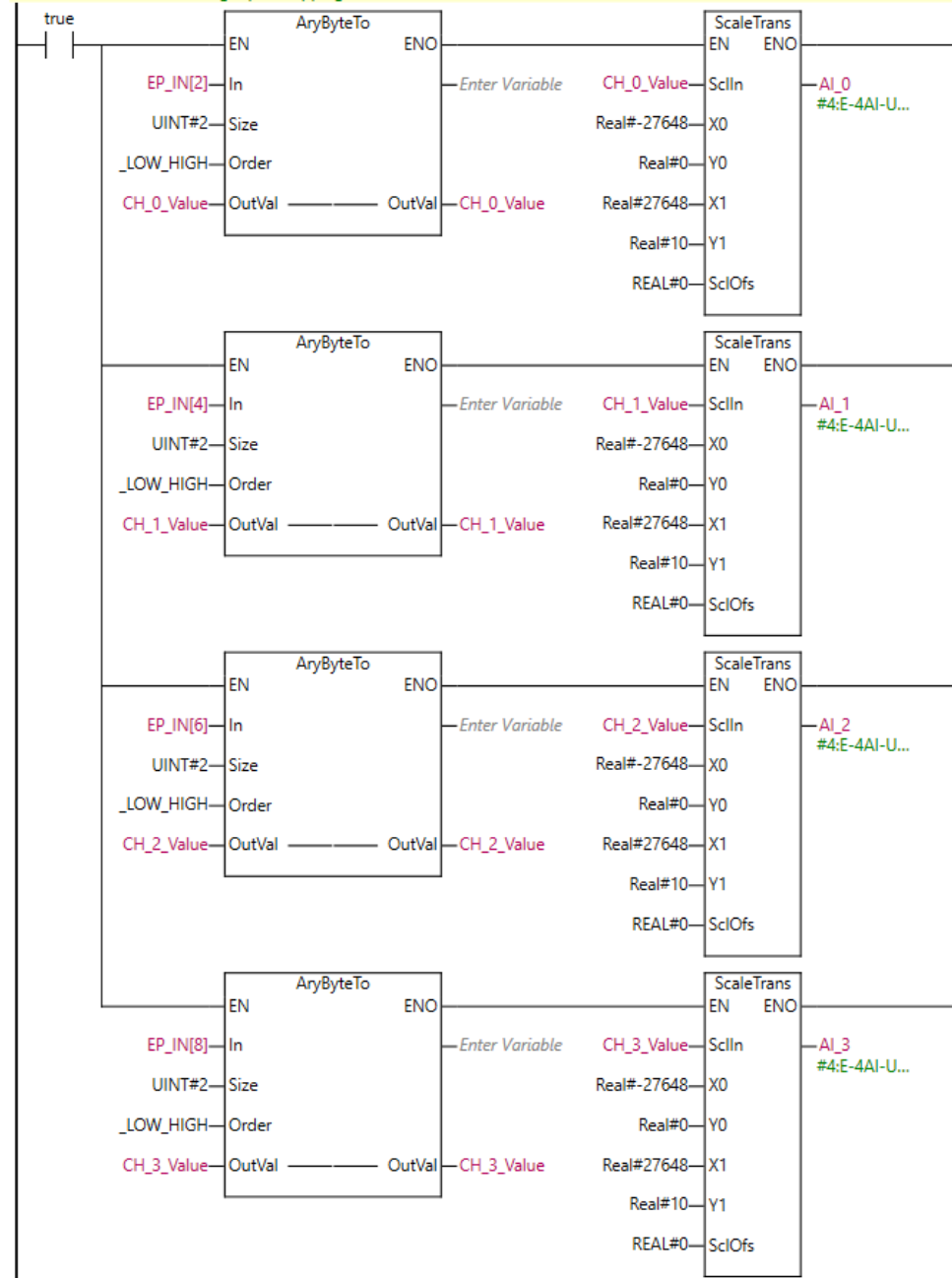
## 6.7 Addressing of Analog Input Modules

Analog channel address is split in to two different bytes. So user needs to convert 2 bytes into one real format. To find the correct byte address of input channels see the description of input array & to find the correct byte address of input channels see the description of input array. Below illustration gives idea to find byte addresses of input modules.



## 6.8 Sample Code for Analog Input Module

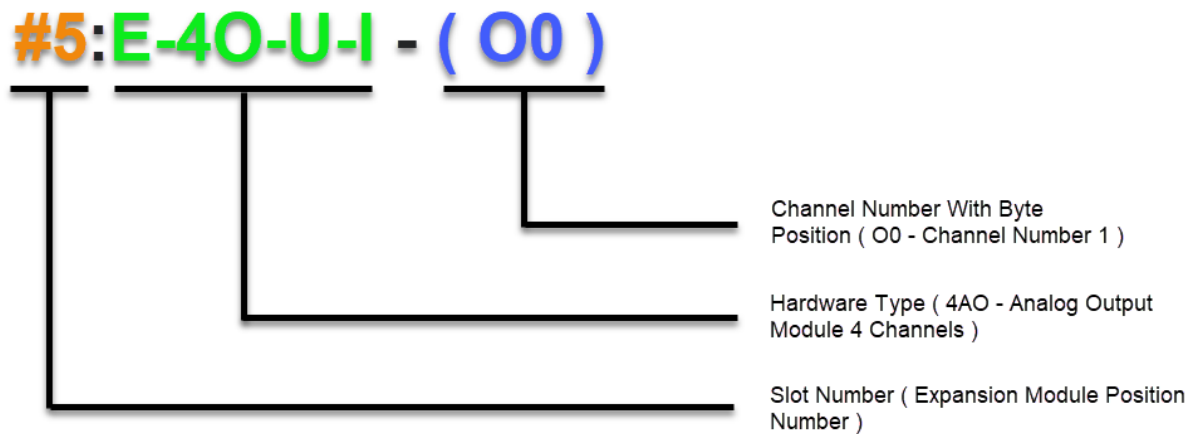
Slot - 4 CPX-E-4AI-U-I Analog Input Mapping From AI0 to AI3



## 6.9 Addressing of Analog Output Modules

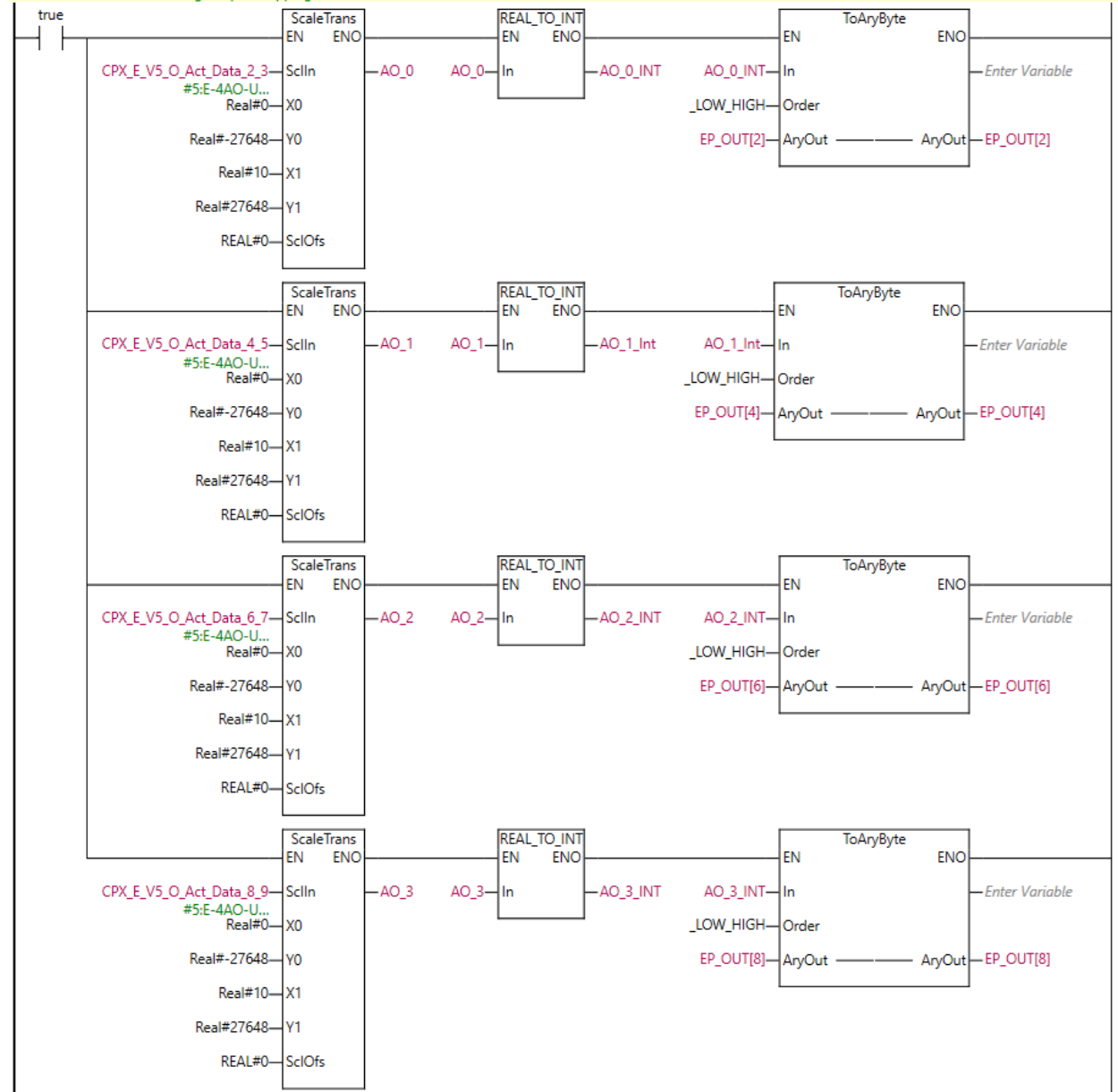
Analog channel address is split in to two different bytes. So user needs to convert 2 bytes into one integer format. To find the correct byte address of input channels see the description of input array & to find the correct

byte address of output channels see the description of output array. Below illustration gives idea to find byte addresses of output modules.



## 6.10 Sample Code for Analog Output Module

Slot - 5 CPX-E-4AO-U-I Analog Output Mapping From AO0 to AO3



## 7 Diagnosis CPX-E-EP Module Errors

Diagnostic information for the automation system CPX-E is made available via the I/O diagnostic interface. The I/O diagnostic interface enables bus independent read access to diagnostic information, data and parameters via 16 internal inputs and outputs.

Various options are available for diagnosing errors depending on the parameterization and network protocol used.

| Diagnosics option                     | Description                                                                                                                                                                                                                                                    |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LED display components                |                                                                                                                                                                                                                                                                |
| System-specific                       | The system status and errors are displayed directly on the module via LED indicators.                                                                                                                                                                          |
| Module- or network-specific           | Module or network status and errors are displayed directly on the module via LED indicators.                                                                                                                                                                   |
| Internal system diagnostics           |                                                                                                                                                                                                                                                                |
| Status bits                           | Common diagnostic messages (global error messages) are displayed by 8 internal inputs (8 status bits).                                                                                                                                                         |
| I/O diagnostics interface             | Diagnostic information for the automation system CPX-E is made available via the I/O diagnostic interface. The I/O diagnostic interface enables bus-independent read access to diagnostic information, data and parameters via 16 internal inputs and outputs. |
| Network-specific diagnostic functions | Special diagnostic functions or communication services may be available, such as DPV1 for PROFIBUS depending on the network used.                                                                                                                              |

Tab. 12 Diagnostics options

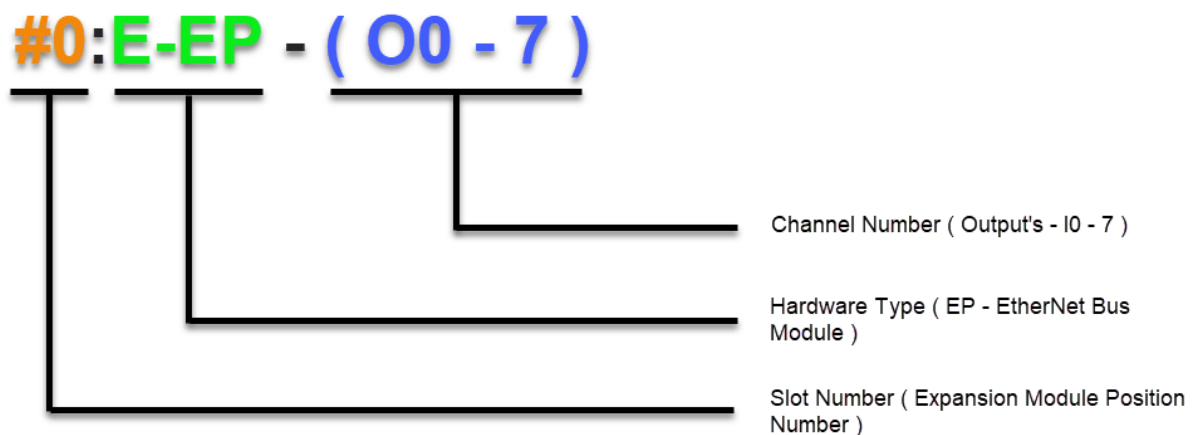
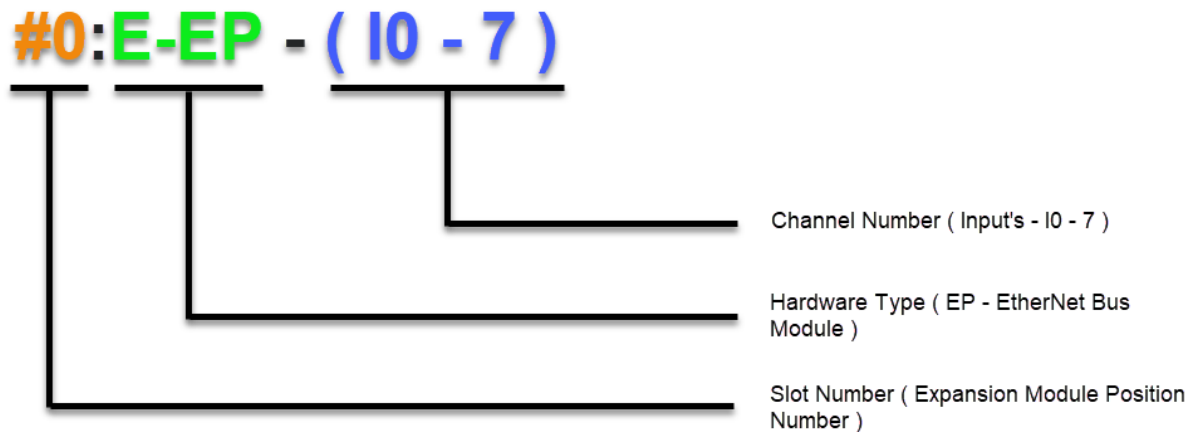


### Note

- Please refer chapter 3.4 for selection of diagnostics option.

## 7.1 Addressing of Diagnostics Data

To find the correct bit address of input channels see the description of input tag array & to find the correct bit address of input channels see the description of input tag array. Below illustration gives idea to find bit addresses of input module's.



## 7.2 I/O Diagnostics Interface

For networks that do not have extensive diagnostic functions, the diagnostic information for the automation system CPX-E is available via the I/O diagnostic interface. The I/O diagnostic interface enables network-independent read access to diagnostic information, data and parameters via internal inputs and outputs (16 I/16 O).

## 7.3 Organisation of Internal Data and Parameters

Internal data and parameters of the modules and of the automation system CPX-E are stored in a common memory area. The I/O diagnostic interface can provide read access to individual bytes of this memory area with the aid of the function number.

7.4 Mode of Operation

Detailed diagnostic information can be accessed via the I/O diagnostic interface.

For example, it is possible to ascertain on exactly which module and which channel an error has occurred.

16 input bits and 16 output bits, through which all diagnostic data can be retrieved, are available for accessing the system diagnostics.



**Note**

- The addresses of the input and output bits of the I/O diagnostic interface depend on the network used.

7.4.1 Output Bits

The function number of the desired data is specified in binary coded form using the output bits.

A0 ... A12 of the I/O diagnostic interface. The function number is accepted if control bit O15 returns a 1-signal.

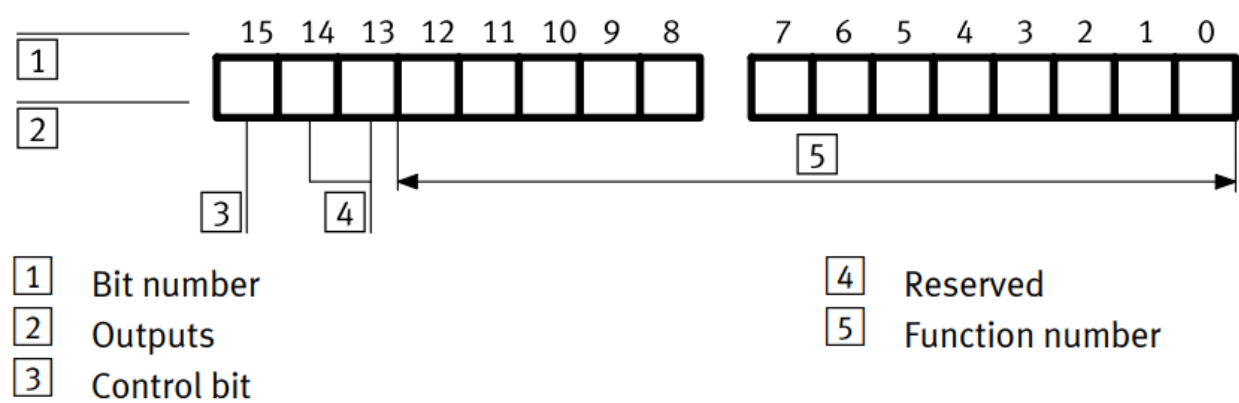


Fig. 14 Output bits

7.4.2 Input Bits

The reply data are output by the automation system CPX-E via the input bits E0 ... E7 when acknowledgement bit E15 returns a 1-signal.

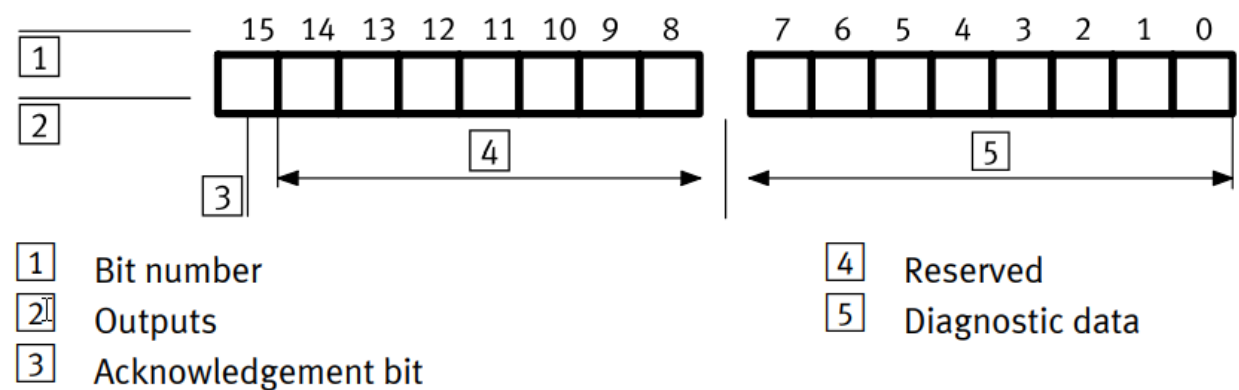


Fig. 15 Input bits

**Note**

- If control bit O15 supplies a logic 0, acknowledgement bit I15 will be reset automatically and the status byte will be shown with the diagnostic data bits.

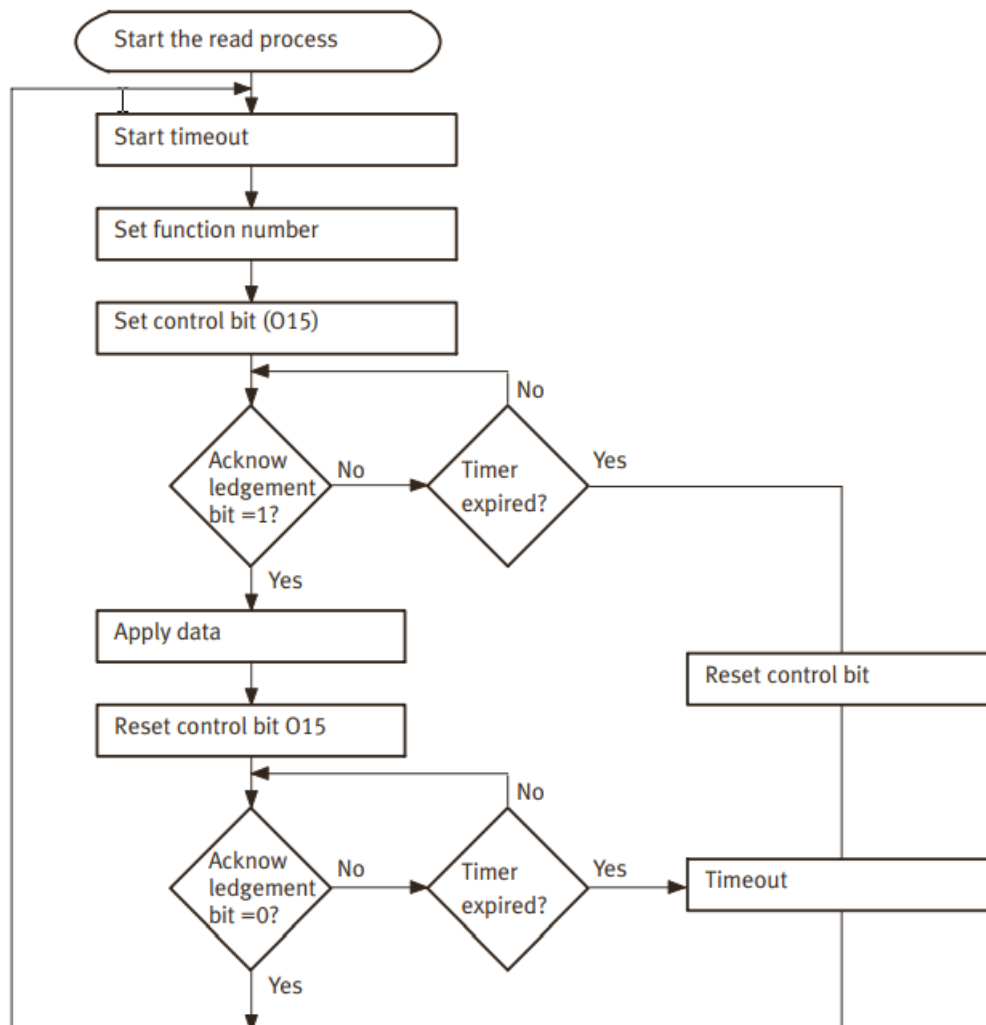
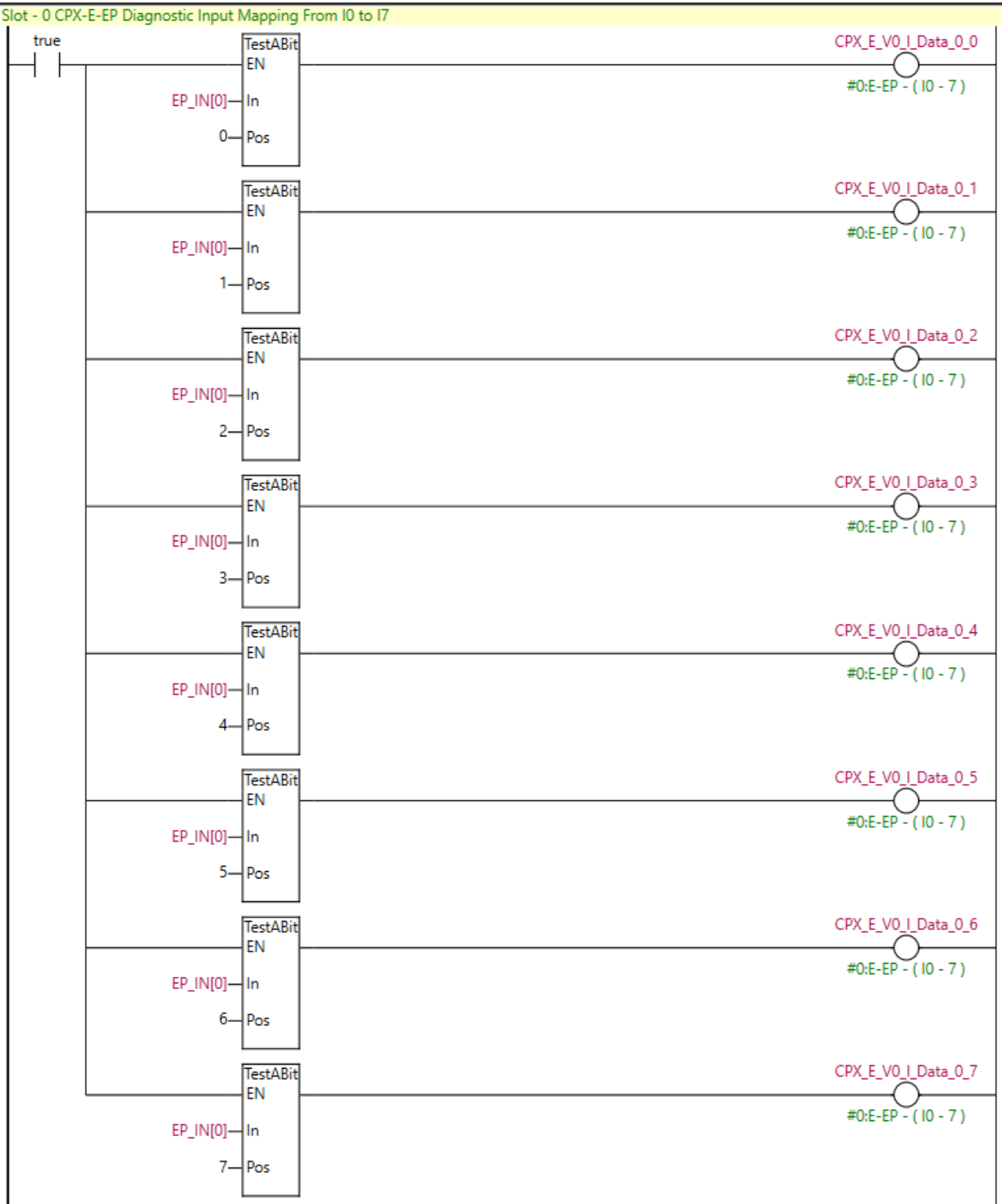
**7.4.3 Reading Out Diagnostic Data**

Fig. 16 Reading out the diagnostic data flow diagram

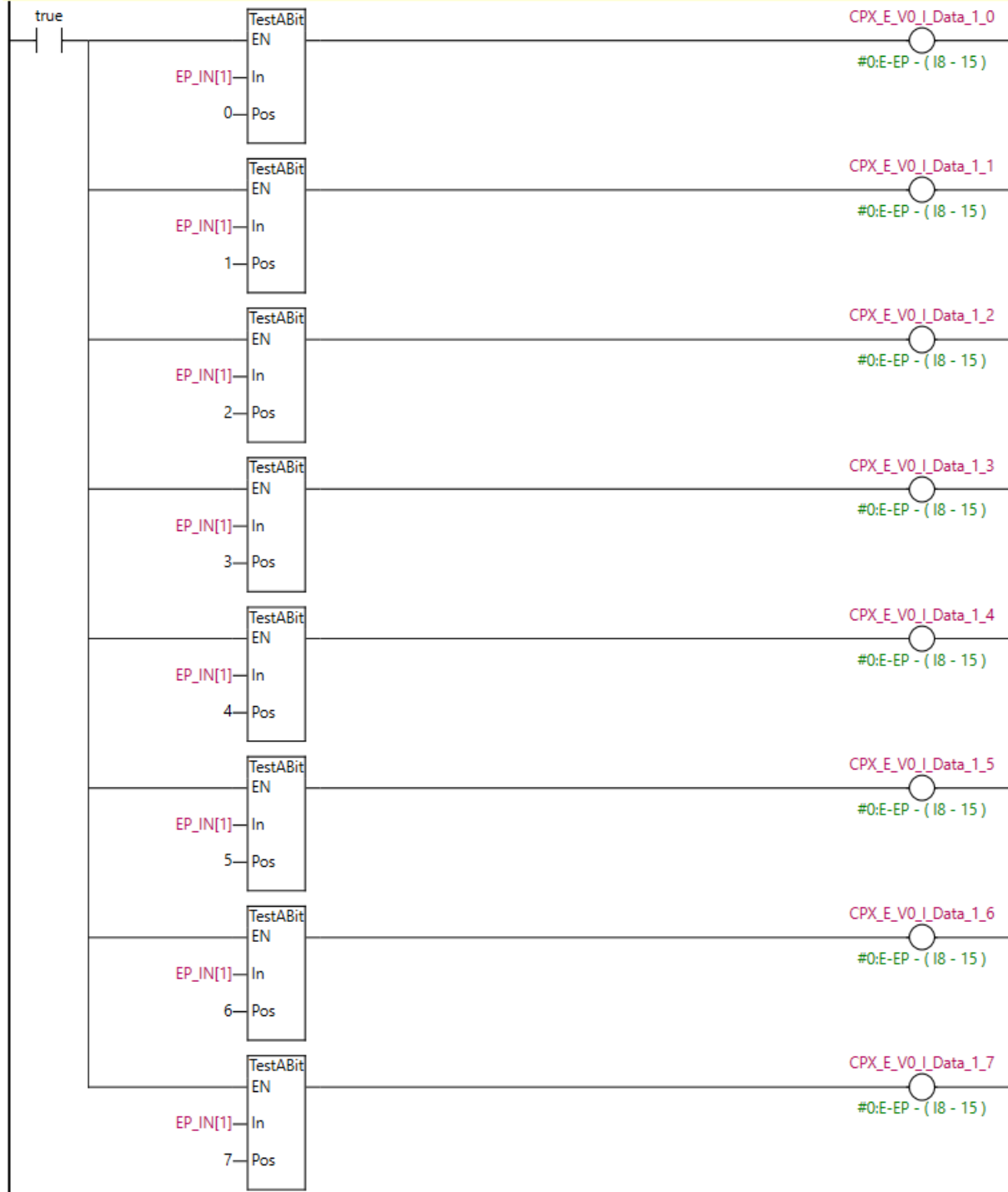
The function number is applied if there is a rising edge at control bit O15.

Input bits I0 ... I7 supply the diagnostic data when the acknowledgement bit supplies a logic 1.

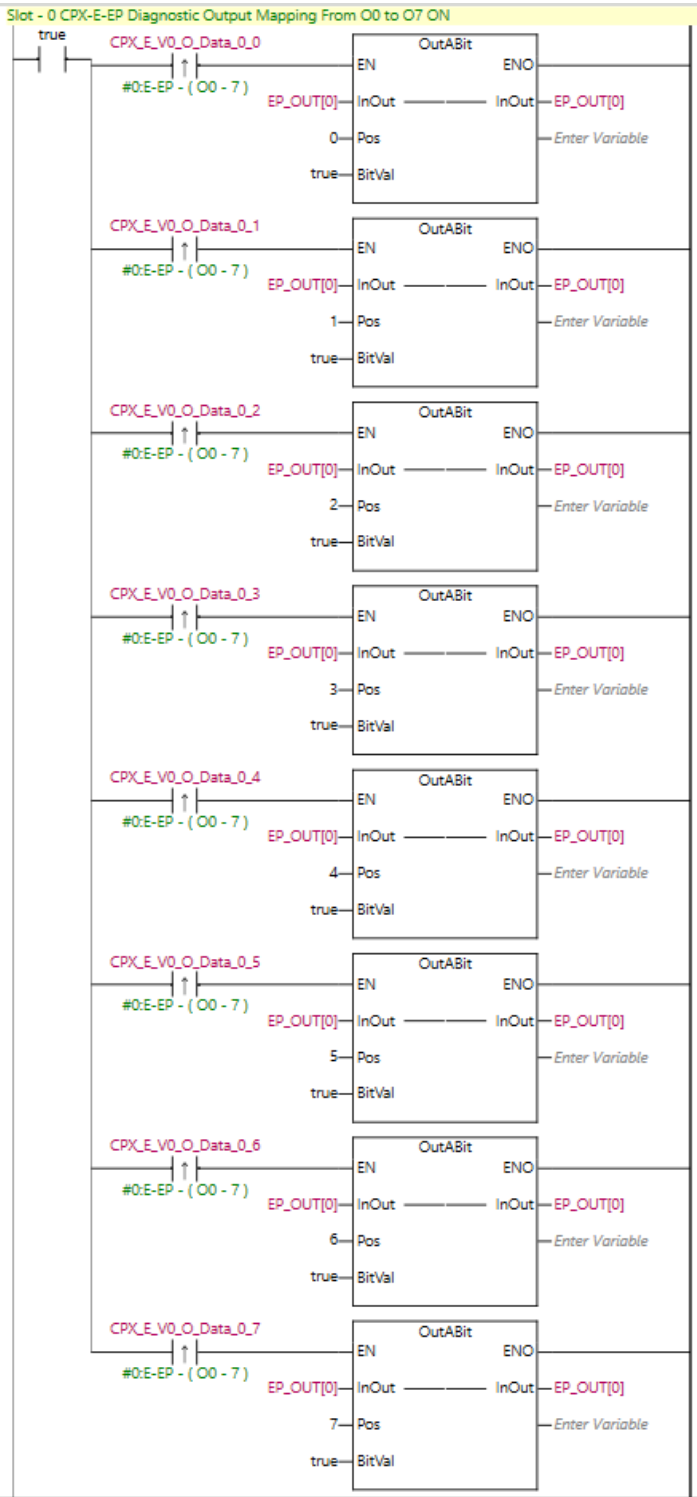
7.5 Sample Code for Diagnostics Data Input

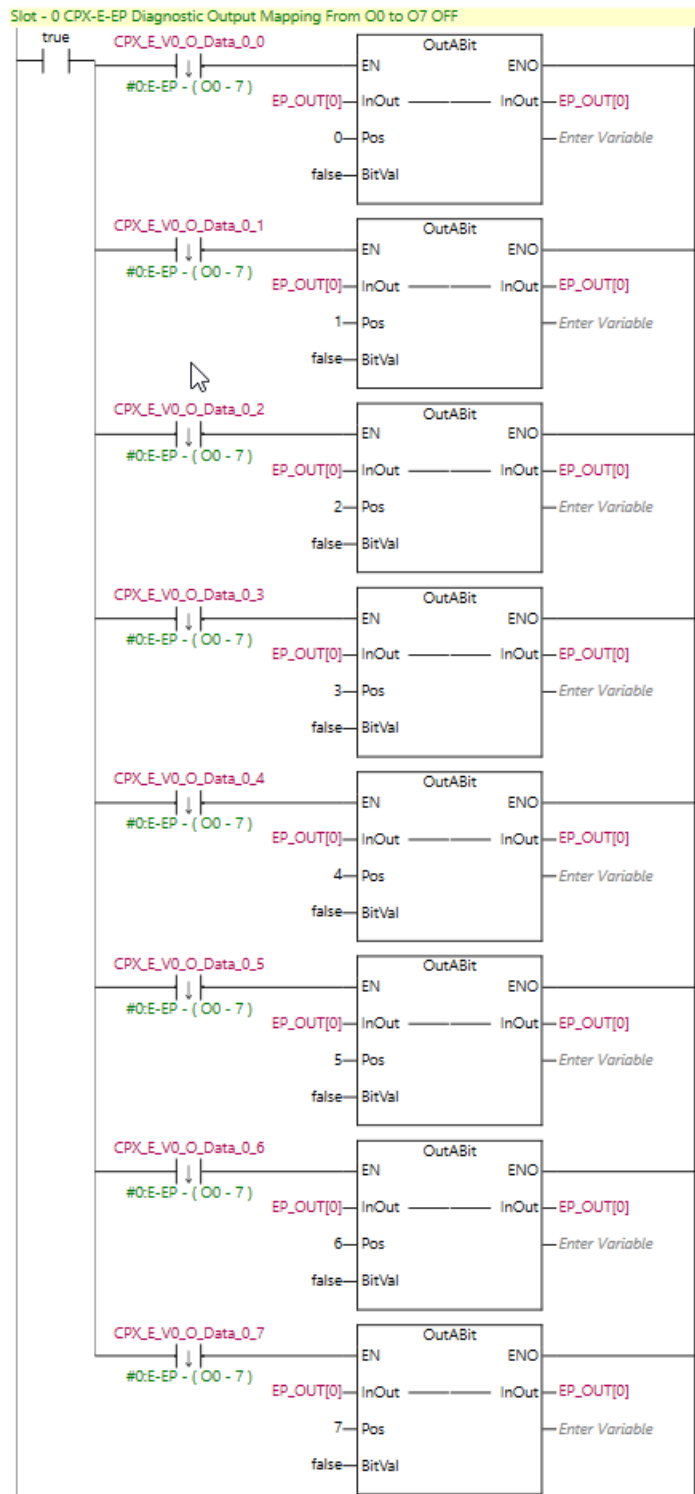


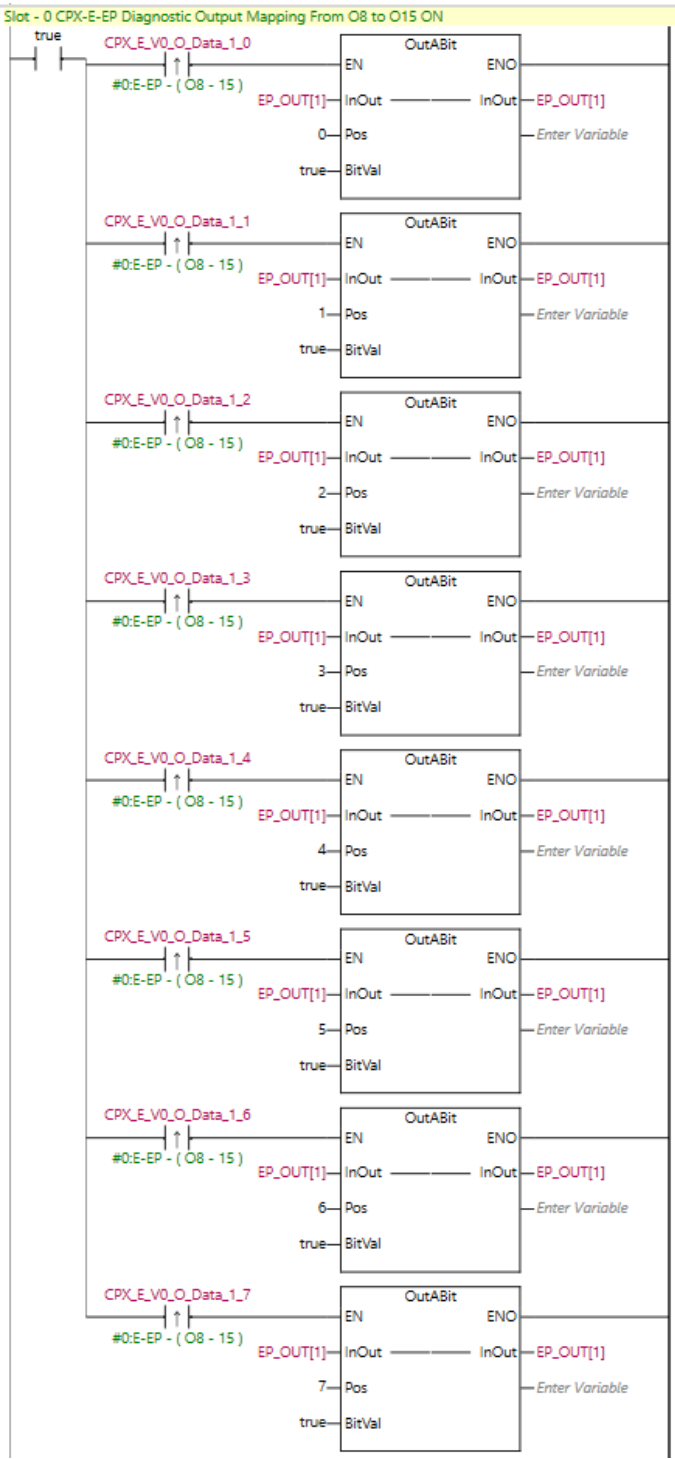
Slot - 0 CPX-E-EP Diagnostic Input Mapping From I8 to I15

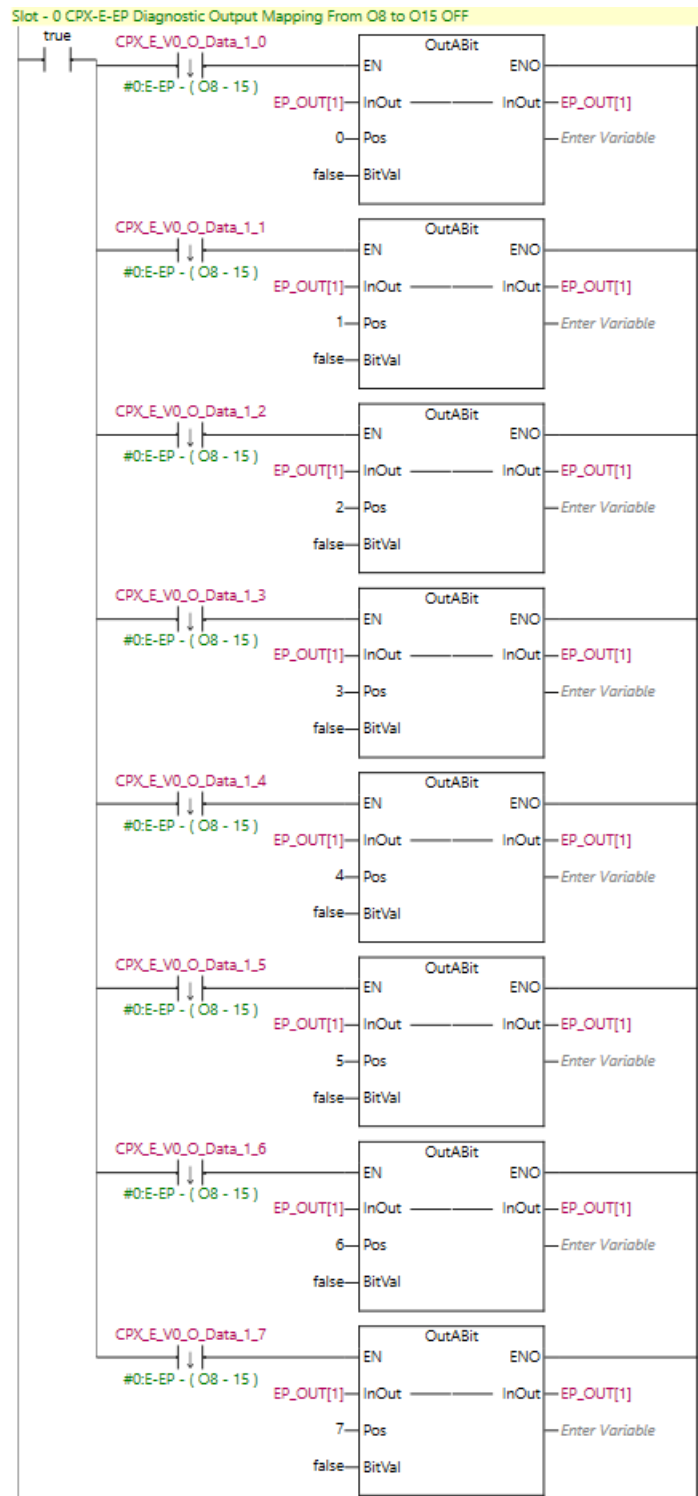


7.6 Sample Code for Diagnostics Data Output









## Note

- Please refer chapter **2.3 Diagnostics Option** from manual CPX-E Automation System – Description – Function – Parameterization for more details.
- Link has been provided on chapter 7.

## 8 Help

Here are some web-linked technical manuals available that related to this topics:

- CPX-E-EP Instruction Manual ---> [CPX-E-EP Instruction](#)
- CPX-E-EP System Manual ---> [CPX-E-EP Description](#)
- CPX-E-EP Bus Node Manual ---> [CPX-E-EP Function](#)
- CPX-E-4IOL Instruction Manual ---> [CPX-E-4IOL Instruction](#)
- CPX-E-4IOL Parameterization ---> [CPX-E-4IOL Description](#)
- CPX-E-16DI Instruction Manual ---> [CPX-E-16DI Instruction](#)
- CPX-E-16DI Parameterization ---> [CPX-E-16DI Description](#)
- CPX-E-8DO Instruction Manual ---> [CPX-E-8DO Instruction](#)
- CPX-E-8DO Parameterization ---> [CPX-E-8DO Description](#)
- CPX-E-4AI-U-I Instruction Manual ---> [CPX-E-4AI-U-I Instruction](#)
- CPX-E-4AI-U-I Parameterization ---> [CPX-E-4AI-U-I Description](#)
- CPX-E-4 AO-U-I Instruction Manual ---> [CPX-E-4AO-U-I Instruction](#)
- CPX-E-4 AO-U-I Parameterization ---> [CPX-E-4AO-U-I Description](#)
- Festo Field Device Tool S/W ---> [Festo FDT](#)
- CPX Festo Maintenance Tool S/W ---> [Festo CPX-FMT](#)
- Festo Automation Suite S/W ---> [Festo FAS](#)
- Sysmac Studio User Manual ---> [Sysmac User Manual](#)
- Omron NJ/NX CPU Manual ---> [NJ/NX CPU Software Manual](#)
- Omron NJ/NX CPU H/W Manual ---> [NJ/NX CPU Hardware Manual](#)
- Omron NJ/NX CPU Ethernet Port Manual ---> [NJ/NX CPU Built-in EetherNet Port Manual](#)